

Submitted to ApJS on July 6, 2011

AZEuS: An Adaptive Zone Eulerian Scheme for Computational MHD

Jon P. Ramsey¹, David A. Clarke

Institute for Computational Astrophysics, Department of Astronomy & Physics, Saint Mary's University, Halifax, Nova Scotia B3H 3C3, Canada.

and

Alexander B. Men'shchikov

Laboratoire AIM, CEA/DSM-CNRS-Université Paris Diderot, IRFU/SAP, CEA Saclay, 91191 Gif-sur-Yvette, France.

ABSTRACT

A new adaptive mesh refinement (AMR) version of the *ZEUS-3D* astrophysical magnetohydrodynamical (MHD) fluid code, **AZEuS**, is described. The AMR module in **AZEuS** has been completely adapted to the staggered mesh that characterises the *ZEUS* family of codes, on which scalar quantities are zone-centred and vector components are face-centred. In addition, for applications using static grids, it is necessary to use higher-order interpolations for prolongation to minimise the errors caused by waves crossing from a grid of one resolution to another. Finally, solutions to test problems in 1-, 2-, and 3-dimensions in both Cartesian and spherical coordinates are presented.

Subject headings: hydrodynamics — magnetohydrodynamics (MHD) — methods: numerical

¹Current address: Zentrum für Astronomie der Universität Heidelberg, Institut für Theoretische Astrophysik, Albert-Ueberle-Str. 2, 69120 Heidelberg, Germany.

1. Introduction

High-resolution, multidimensional simulations have become indispensable for many complex problems in astrophysics, particularly those involving (magneto-)fluid dynamics. One of the most important innovations in this area has been the use of dynamic and variable resolution techniques. Adaptive mesh refinement (AMR), pioneered in the context of the fluid equations by [Berger & Olinger \(1984\)](#) and [Berger & Colella \(1989; BC89\)](#), is one such approach.

With AMR, a hierarchy of grids is used to provide high numerical resolution when and where the physics requires it, leaving as much of the volume at lower resolution as possible to minimise computational effort. This makes AMR an efficient means of studying problems with a very large spatial dynamic range (*e.g.*, star formation, galaxy evolution), as borne out by the large number of codes which employ it: *ORION* ([Klein, 1999](#)), *FLASH* ([Fryxell *et al.*, 2000](#)), *RIEMANN* ([Balsara, 2001](#)), *RAMSES* ([Fromang *et al.*, 2006](#)), *PLUTO* ([Mignone *et al.*, 2007](#)), *NIRVANA* ([Ziegler, 2008](#)), *AstroBEAR* ([Cunningham *et al.*, 2009](#)), and *ENZO* ([Collins *et al.*, 2010](#)) to name several.

Virtually all AMR fluid codes to date are based on a zone-centred grid, with all hydrodynamical variables (density, energy, and momentum components) taken to be located at the centres of their respective zones. Indeed, AMR was originally designed specifically for zone-centred schemes. Magnetohydrodynamic (MHD) solvers are designed with either zone-centred or face-centred magnetic field components, depending in part on the mechanism used to preserve the solenoidal condition. One scheme that has enjoyed somewhat of a renaissance of late is Constrained Transport (CT; [Evans & Hawley 1988](#)), which places magnetic field components at the centres of the zone-faces to which they are normal. The staggered mesh introduced in such a scheme has to be specifically accounted for in the AMR modules and in such a way that $\nabla \cdot \vec{B}$ remains zero everywhere—including within the boundaries—to machine round-off error.

The *ZEUS* family of codes are one of only a very few astrophysical fluid codes in use that employ a *fully* staggered grid (*e.g.*, *STAGGER*; [Kritsuk *et al.* 2011](#) and references therein), where the momentum components are also face-centred (Figure 1). While concerns have been expressed over the suitability of its MHD algorithms in certain pedagogical 1-D test problems (*e.g.*, [Falle 2002](#)), the fact remains that in one form or another, *ZEUS* is among the best tested, documented, and widely-used fluid codes in astrophysics ([Stone & Norman 1992a,b](#); [Stone, Mihalas, & Norman 1992](#); [Clarke 1996, 2007, 2010](#); [Hayes *et al.* 2006](#)), and a proper merger with AMR is warranted. To do this, AMR has to be modified for a fully staggered grid, including the proper treatment of face-centred magnetic field and face-centred momentum.

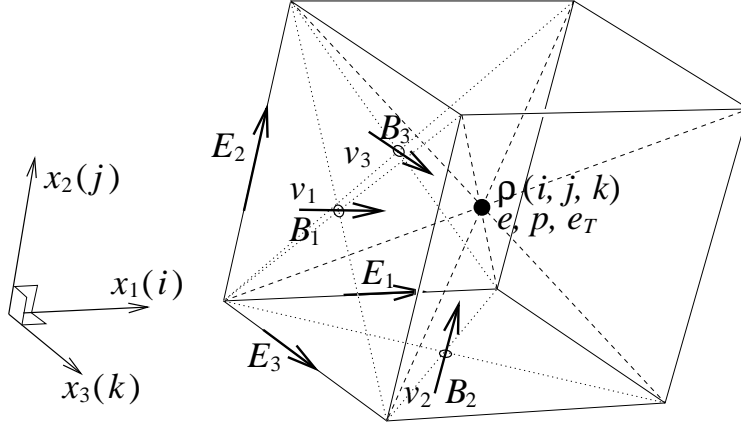


Fig. 1.— Depiction of a single zone on a fully-staggered grid, where scalars (ρ , e_T , e , p) are zone-centred, primitive vectors (\vec{v} , \vec{B}) are face-centred, and derived vectors ($\vec{E} = -\vec{v} \times \vec{B}$, $\vec{J} = \nabla \times \vec{B}$) are edge-centred.

In this paper, we introduce the newest member of the *ZEUS* family of codes, *AZEuS*, whose “maiden simulations” have already appeared in Ramsey & Clarke (2011). *AZEuS* is a block-structured AMR version of *ZEUS-3D* (Clarke, 1996, 2010) which preserves the modularity and structure of the underlying *ZEUS* module. The AMR scheme of BC89, including the changes described in Bell *et al.* (1994), are modified for a fully-staggered grid with additional modifications made to the prolongation procedure to allow for smooth passage of all types of waves between adjacent grids of differing resolution. *AZEuS* is currently capable of ideal MHD in 1-, 1.5-, 2-, 2.5-, and 3-D in Cartesian, cylindrical, and spherical polar coordinates using both dynamic and static grids, and with a full suite of physical boundary conditions. As with all *ZEUS*-type codes, its operator-split design allows for additional physics (*e.g.*, gravity, viscosity, radiation, *etc.*) to be added without concern over how such additions will affect the MHD algorithm. How the non-hyperbolic additions are implemented for an adaptive grid is another matter (*e.g.*, radiation; Wise & Abel 2011).

This paper does not attempt to give a full recount of the basic methodology in either AMR or *ZEUS*, but focuses instead on the modifications to AMR (not so much to *ZEUS*) necessary for their merger. Thus, the reader should be familiar with BC89 and Clarke (1996, 2010). In Section 2, we list the MHD equations solved by *AZEuS*, and define our conventions and notation. In Sections 3 and 4, we enumerate the modifications necessary for restriction and prolongation on a staggered grid, as well as outline the interpolation schemes used to allow the smooth passage of waves across grid boundaries. Section 5 focuses on boundary conditions, while in Section 6 we discuss how grids are created and how the proper nesting

criterion must be modified for a fully-staggered grid. Several of the 1-D, 2-D, and 3-D test problems used to validate AZEuS are given in Section 7, followed by a quick summary in Section 8. Discussion of curvilinear coordinates, use of the vector potential, and a schematic overview of the code are relegated to the appendices.

2. Preamble

2.1. Underlying numerical method

AZEuS solves the following equations of ideal MHD (with the artificial viscosity and gravity terms included):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0; \quad (1)$$

$$\frac{\partial \vec{s}}{\partial t} + \nabla \cdot (\vec{s} \vec{v}) = -\nabla p + (\nabla \times \vec{B}) \times \vec{B} - \nabla \cdot \mathbf{Q} - \rho \nabla \phi; \quad (2)$$

$$\frac{\partial \vec{B}}{\partial t} + \nabla \times \vec{E} = 0; \quad (3)$$

$$\frac{\partial e}{\partial t} + \nabla \cdot (e \vec{v}) = -p \nabla \cdot \vec{v} - \mathbf{Q} : \nabla \vec{v}; \quad (4)$$

$$\frac{\partial e_{\text{T}}}{\partial t} + \nabla \cdot (e_{\text{H}} \vec{v} + \vec{E} \times \vec{B} + \mathbf{Q} \cdot \vec{v}) = 0, \quad (5)$$

where ρ is the mass density, \vec{v} is the velocity, $\vec{s} = \rho \vec{v}$ is the momentum density, p is the thermal pressure, \vec{B} is the magnetic induction¹, \mathbf{Q} is the von Neumann-Richtmyer artificial viscous stress tensor (von Neumann & Richtmyer, 1950; Clarke, 2010), ϕ is the gravitational potential and satisfies the Poisson equation ($\nabla^2 \phi = 4\pi G \rho$), $\vec{E} = -\vec{v} \times \vec{B}$ is the induced electric field, e is the internal energy density, $e_{\text{H}} = e + \frac{1}{2} \rho v^2 + \rho \phi$ is the hydrodynamical energy density, and $e_{\text{T}} = e_{\text{H}} + \frac{1}{2} B^2$ is the total energy density. This set of equations [in which equation (5) can be derived from equations (1)–(4)] is closed by the ideal gas law, $p = (\gamma - 1)e$, where γ is the ratio of specific heats. Figure 1 shows the locations of most of these variables on a fully-staggered grid. Other physics terms often found in ZEUS codes such as a second fluid, physical viscosity, radiation, *etc.*, have yet to be implemented.

AZEuS inherits the operator-split methodology of ZEUS-3D, wherein the terms on the RHS of equations (1)–(5) are treated in a *source step* while those on the LHS are accounted for in a separate *transport* and *inductive step*. As such, the algorithm is not strictly conservative.

¹in units where $\mu_0 = 1$

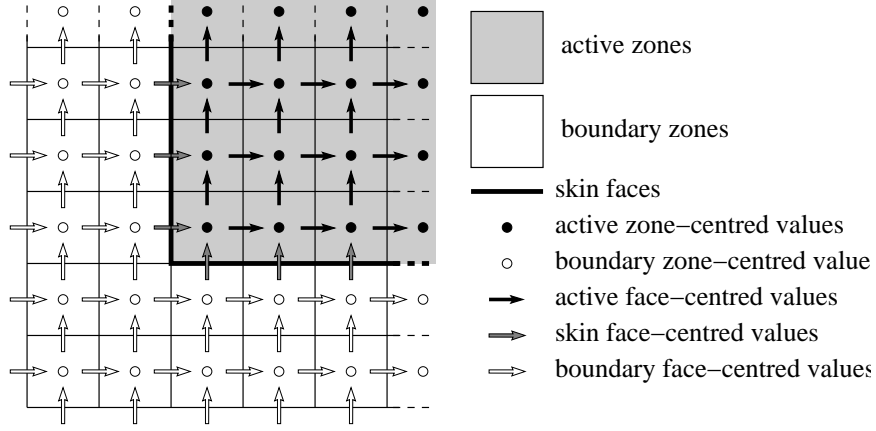


Fig. 2.— On a fully-staggered grid, all variables have two boundary values. In addition, for each direction, one component of a face-centred vector and two components of an edge-centred vector have one skin value.

However, based as it is on the version of *ZEUS-3D* described by [Clarke \(2010\)](#), *AZEuS* can solve either the internal energy equation or the total energy equation, where the latter choice does ensure conservation of total energy to machine round-off error, but at the cost of non-positive-definite thermal pressure. Should positive definite pressures be paramount, the internal energy equation offers a viable option with, in most cases, total energy conserved to within 1% or less (see [Clarke 2010](#) for further discussion).

To accommodate the interpolation schemes, two boundary zones must be specified at the edges of all grids. On a staggered grid, all zone-centred quantities have just the two boundary values, while face-centred quantities have two boundary values plus a value that lies on the face separating the “active zones” from the “boundary zones”, henceforth referred to as the “skin” of the grid (Figure 2). As we shall see, skin values for the magnetic field are treated just like active zones, while the momenta on the skin are treated somewhere in between active and boundary values; the difference attributed to the conservation properties of these two quantities.

2.2. Conventions and Notation

We adopt the following conventions and notation throughout this paper:

1. Quantities in coarse and fine zones are denoted with upper and lower cases respectively: *e.g.*, $Q(I, J, K)$, $q(i, j, k)$. Fluxes for a quantity Q (q) are denoted $F_{m,Q}$ ($f_{m,q}$), where

$m = 1, 2$, or 3 indicates the component direction.

2. If a “coarse” grid or zone is considered to be at level l , its daughter “fine” grid or zone is at level $l + 1$. The base and coarsest grid, which covers the entire domain, is at level $l = 1$. The refinement ratio, ν , between level l and $l + 1$ must be a power of 2 and the same in all directions.
3. Grid volumes, areas, lengths and time steps are ΔV , ΔA_m , Δx_m , and Δt for a coarse grid, and δV , δA_m , δx_m , and δt for a fine grid.
4. Indices (i, j, k) correspond to the fine zone at the (left, bottom, back) of a coarse zone with indices (I, J, K) . The zone-centre of a particular fine zone within a coarse zone is designated $(i + \alpha, j + \beta, k + \eta)$, where $\alpha, \beta, \eta = 0, 1, \dots, \nu - 1$. For the 1-face-centre of a fine zone, $\alpha = 0, 1, \dots, \nu$ while $\beta, \eta = 0, 1, \dots, \nu - 1$, *etc.*
5. Similarly, grid positions for the coarse grid use upper case indices [*e.g.*, $x_1(I)$], while grid positions for the fine grid use lower-case indices [*e.g.*, $x_1(i)$].
6. The current time step for the coarse grid is indicated by the upper case superscript N , while the current fine time step is indicated by the lower case superscript n . Typically, $n = \nu N$. To designate a fine time step within a coarse time step, we use $n + \tau$ where $0 \leq \tau \leq \nu - 1$ ².
7. The *region of influence* (ROI) of a variable is defined as the area or volume over which that quantity is conserved. For zone-centred scalars on the coarse grid, this is the volume $\Delta V(I, J, K)$ (Figure 3a). For face-centred but volume-conserved quantities (*e.g.*, S_1), the ROI is the *staggered* volume [*e.g.*, $\frac{1}{2}(\Delta V(I, J, K) + \Delta V(I - 1, J, K))$] (Figure 3b). Finally, for face-centred but area-conserved quantities (*e.g.*, B_1), the ROI is the area of the face at which the vector component is centred [*e.g.*, $\Delta A_1(I, J, K)$] (Figure 3c).

Finally, while AZEuS is written in the covariant fashion of *ZEUS-2D* (Stone & Norman, 1992a), our discussion is given in terms of Cartesian-like components with uniform zone sizes within each grid for simplicity. As such, $\Delta V/\delta V = \nu^3$, $\Delta A_m/\delta A_m = \nu^2$, $\Delta x_m/\delta x_m = \nu$, and $\Delta t/\delta t = \nu$. Some of the modifications necessary for curvilinear coordinates are given in Appendix A.

²While a coarse zone is always divided in each direction into ν fine zones, a coarse time step isn’t necessarily divided into ν fine time steps; additional fine time steps are taken if local CFL conditions demand it. For simplicity of description, however, we shall proceed as though $n = \nu N$, though the reader should be aware that this may not always be true.

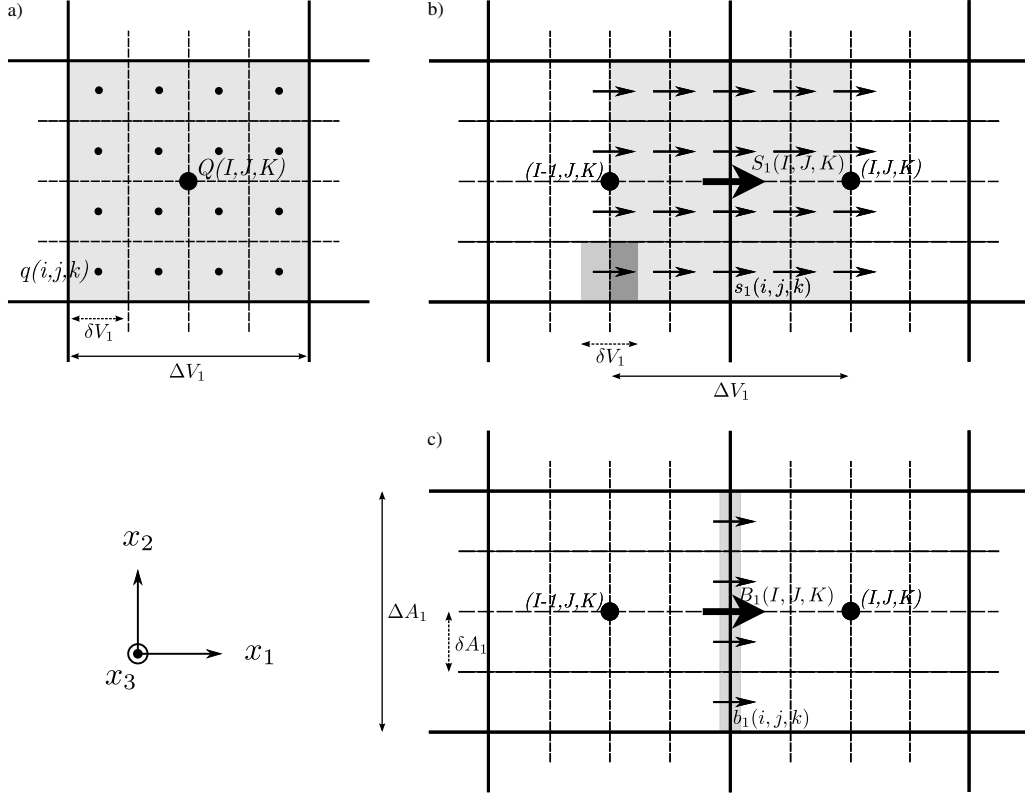


Fig. 3.— The regions of influence (ROI) (shaded) for: (a) zone-centred variables, (b) face-centred and volume-conserved variables, and (c) face-centred and area-conserved variables. A refinement ratio of $\nu = 4$ is shown.

3. Restriction

Restriction is the process by which data on the coarse grid are replaced by an average of data from an overlying fine grid. This must be done in a fashion that locally preserves all conservation laws and the solenoidal condition to within machine round-off error. Two types of restriction are considered: the conservative overwrite of coarse values with ROIs which are entirely covered by ROIs of overlying fine zones, and flux corrections to coarse zones (sometimes called “refluxing”) with ROIs which are adjacent to, or partially covered by, the ROIs of fine zones.

3.1. Conservative Overwrite

At the end of a coarse time step, fine and coarse grids are synchronised by overwriting the coarse grid with “better” values from overlying fine grids. Because of the different ROIs on a staggered mesh, the specifics of the overwriting procedure depend on which variable is being overwritten. For zone-centred, volume-conserved quantities (*e.g.*, ρ , e , e_T), the procedure is the same as in [BC89](#):

$$Q(I, J, K) = \frac{1}{\nu^3} \sum_{\alpha, \beta, \eta=0}^{\nu-1} q(i + \alpha, j + \beta, k + \eta), \quad (6)$$

where the sum is a triple sum. By inspection, one can confirm that equation (6) conserves Q (q) locally to within machine round-off error.

For face-centred, volume-conserved quantities such as the momentum density whose ROIs are completely covered by the ROIs of overlying fine zones, we have:

$$S_1(I, J, K) = \frac{1}{\nu^3} \sum_{\alpha=-\nu/2}^{\nu/2} \sum_{\beta, \eta=0}^{\nu-1} \mathcal{G}(\alpha) s_1(i + \alpha, j + \beta, k + \eta), \quad (7)$$

$$\text{where } \mathcal{G}(\alpha) = \begin{cases} 1/2 & \text{if } \alpha = \pm \nu/2 \\ 1 & \text{otherwise,} \end{cases} \quad (8)$$

for the 1-component of the momentum. The factor $\mathcal{G}(\alpha)$ takes into account that only half of the ROIs of the fine momenta at $\alpha = -\nu/2, \nu/2$ cover the ROI of coarse momentum ([Figure 3b](#)).

Coarse momenta with ROIs partially covered by a fine grid are co-spatial with the skin of the overlying fine grid. As skin values of momenta are considered to be boundary values (since one of the fluxes is completely determined from within the boundary), they are not taken to be more reliable than the underlying coarse values (whose fluxes are determined exclusively by interior zones), and thus the coarse values are not overwritten by the fine grid values. Instead, coarse momenta co-spatial with a fine grid skin are considered to be adjacent to the fine grid and, as such, are subject to the “flux-correction” step described in the next subsection.

For magnetic field, the conserved quantity is the magnetic flux ($\int \vec{B} \cdot d\vec{A}$). Thus, coarse values of B_1 are overwritten using:

$$B_1(I, J, K) = \frac{1}{\nu^2} \sum_{\beta, \eta=0}^{\nu-1} b_1(i, j + \beta, k + \eta), \quad (9)$$

where the sum is over all fine ROIs (areas of 1-faces) that cover the coarse ROI. One can easily show that overwritten values of \vec{B} will still satisfy the solenoidal condition—even when combined with values of \vec{B} that are not overwritten—so long as the overlying values of \vec{b} are divergence-free and adjacent values of \vec{B} are properly “refluxed” (Section 3.2). In addition, since \vec{B} is an area-conserved quantity, there is never partial coverage of ROIs as there is with the momenta, and thus no analogue of $\mathcal{G}(\alpha)$ (equation 8) is necessary for the magnetic field.

A straightforward permutation of indices gives the analogous expressions for the other components of momentum and magnetic field.

3.2. Flux Corrections

A coarse zone adjacent to but not covered by a fine grid shares a face with the fine grid. In order that local mass, momentum, and magnetic flux remain conserved to within machine round-off error, the coarse and fine zones must agree on the fluxes passing across their common face. This is accomplished by keeping track of all coarse fluxes passing across the skin of a contained fine grid, and then subtracting from these the fine fluxes computed during the MHD updates of the fine grids. These “flux corrections” are then subtracted from the coarse zones adjacent to the fine grid during the so-called “refluxing step”, effectively replacing the coarse fluxes with the fine fluxes along their common face.

For zone-centred, volume-conservative quantities this procedure follows BC89. Thus, for transport in the 1-direction, we have for the flux-corrected quantity, \tilde{Q} :

$$\begin{aligned} \tilde{Q}^{N+1}(I, J, K) = & Q^{N+1}(I, J, K) - \frac{1}{\Delta V(I, J, K)} \left[F_{1,Q}^{N+\frac{1}{2}}(I, J, K) \right. \\ & \left. - \sum_{\beta, \eta, \tau=0}^{\nu-1} f_{1,q}^{n+\tau+\frac{1}{2}}(i, j+\beta, k+\eta) \right], \end{aligned} \quad (10)$$

where the quantities in square brackets are the flux corrections. For the purpose of illustration, the coarse zone (I, J, K) is taken to be immediately to the right (increasing I) of a fine grid. $F_{1,Q}^{N+1/2}$ is the time-centred 1-flux of Q (with units $Q\mathcal{V}\Delta A\Delta t$, where \mathcal{V} is the coarse velocity)³ passing across the 1-face co-spatial with the skin of the fine grid, while $f_{1,q}^{n+\tau+1/2}$ are the corresponding fine fluxes (with units $qv\delta A\delta t$, where v is the fine velocity; Figure 4a). Note the sum over τ reflecting the fact there are several fine time steps (typically ν of them) within a single coarse time step.

³Strictly speaking, this is not a flux because of the factor Δt . However, for the purposes of accounting, we find it advantageous to define the fluxes with the time steps embedded.

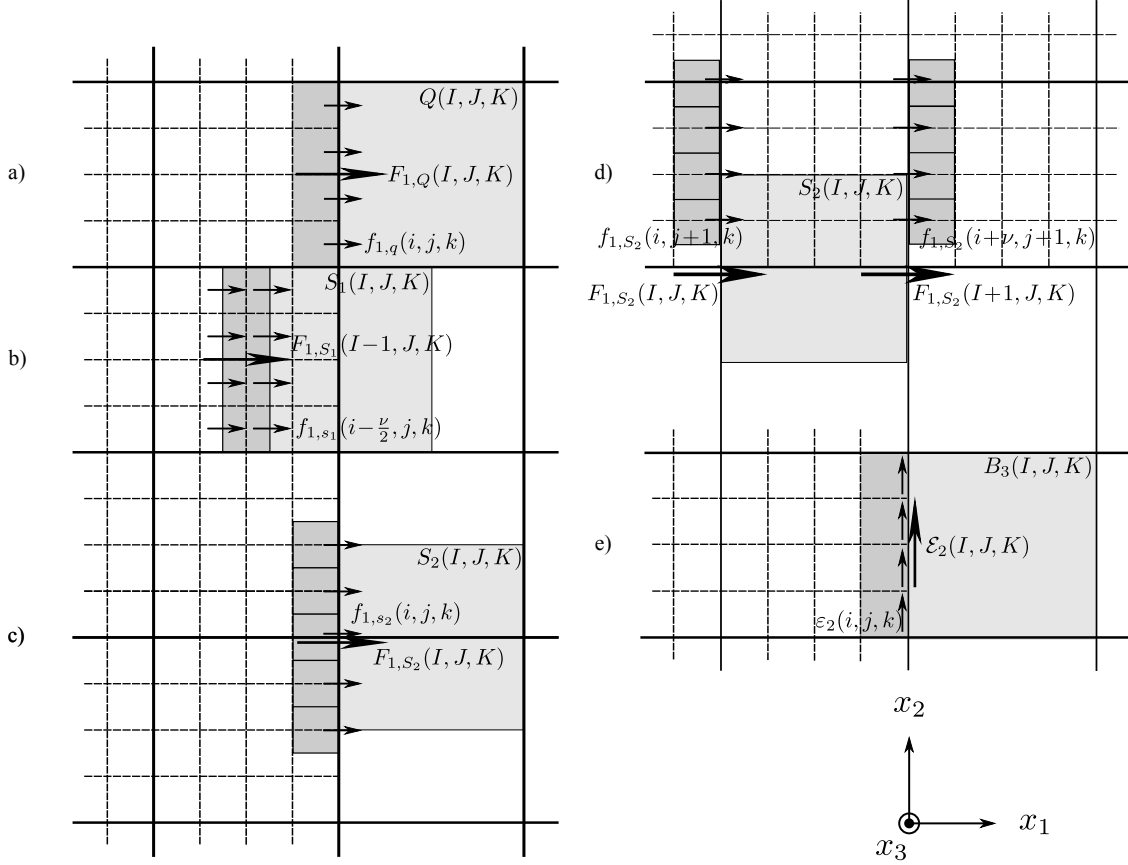


Fig. 4.— The different cases for flux corrections on a staggered grid, including: (a) zone-centred quantities; (b, c, and d) the three different cases for face-centred, volume-conserved momenta; and (e) area-conserved magnetic field corrections via the EMFs. Shaded regions denote typical ROIs referred to in the text. Note that in this figure, all arrows correspond to components of fluxes or EMFs.

When flux correcting the face-centred, volume-conserved momenta, we must depart from BC89 as the staggered mesh gives rise to four distinct cases that must be dealt with individually. The first case is when both the flux and momentum component are parallel to the fine grid skin normal. Here, the ROI at the boundary is halfway inside the fine grid. For example, consider the 1-flux of S_1 in the ROI straddling the right boundary of a fine grid as

shown in Figure 4b where the 1-flux correction takes the form:

$$\begin{aligned} \tilde{S}_1^{N+1}(I, J, K) = & S_1^{N+1}(I, J, K) - \frac{1}{\Delta V(I, J, K)} \left[F_{1, S_1}^{N+\frac{1}{2}}(I-1, J, K) \right. \\ & \left. - \frac{1}{2} \sum_{\beta, \eta, \tau=0}^{\nu-1} \left(f_{1, s_1}^{n+\tau+\frac{1}{2}}(i-\frac{\nu}{2}-1, j+\beta, k+\eta) + f_{1, s_1}^{n+\tau+\frac{1}{2}}(i-\frac{\nu}{2}, j+\beta, k+\eta) \right) \right]. \end{aligned} \quad (11)$$

Note that in the 1-direction, the coarse and fine momenta pass through the same face, but the fluxes do not. Thus, an average of the fine fluxes at $i - \nu/2 - 1$ and $i - \nu/2$ is needed to properly centre the fine fluxes, and to ensure local conservation of momentum.

The second case is when the flux is parallel and the momentum component perpendicular to the fine grid skin normal. Here, the adjacent ROI lies entirely outside the fine grid just as a zone-centred quantity. For example, consider the 1-flux of S_2 in the ROI adjacent the right boundary of a fine grid as shown in Figure 4c, where the flux correction takes the form:

$$\begin{aligned} \tilde{S}_2^{N+1}(I, J, K) = & S_2^{N+1}(I, J, K) - \frac{1}{\Delta V(I, J, K)} \left[F_{1, S_2}^{N+\frac{1}{2}}(I, J, K) \right. \\ & \left. - \sum_{\beta=-\nu/2}^{\nu/2} \sum_{\eta, \tau=0}^{\nu-1} \mathcal{G}(\beta) f_{1, s_2}^{n+\tau+\frac{1}{2}}(i, j+\beta, k+\eta) \right]. \end{aligned} \quad (12)$$

The factor $\mathcal{G}(\beta)$ (equation 8) ensures that only half of the fine 1-fluxes of the fine ROIs at $\beta = \pm\nu/2$ are included, as a quick glance at Figure 4c will verify.

The third case is when the flux is perpendicular and the momentum component parallel to the fine grid skin normal. As with the first case, the ROI at the boundary is halfway inside the fine grid. For example, consider the 1-flux of S_2 in the ROI straddling the lower boundary of a fine grid as shown in Figure 4d, where the flux correction takes the form:

$$\begin{aligned} \tilde{S}_2^{N+1}(I, J, K) = & S_2^{N+1}(I, J, K) \\ & - \frac{1}{\Delta V(I, J, K)} \left[\frac{\nu-1}{2\nu} \left(F_{1, S_2}^{N+\frac{1}{2}}(I, J, K) - F_{1, S_2}^{N+\frac{1}{2}}(I+1, J, K) \right) \right. \\ & \left. - \sum_{\beta=1}^{\nu/2} \sum_{\eta, \tau=0}^{\nu-1} \mathcal{G}(\beta) \left(f_{1, s_2}^{n+\tau+\frac{1}{2}}(i, j+\beta, k+\eta) - f_{1, s_2}^{n+\tau+\frac{1}{2}}(i+\nu, j+\beta, k+\eta) \right) \right]. \end{aligned} \quad (13)$$

The fraction $(\nu-1)/2\nu$ is the *area filling ratio* between the fine to coarse fluxes. For example, for $\nu = 4$, $(\nu-1)/2\nu = 3/8$ (Figure 4d), and 3/8 of the coarse flux must be replaced with the overlapping fine fluxes. Note that $f_{1, s_2}(i, j, k)$ does not contribute to the fine fluxes that replace part of the coarse flux since it is determined, in part, by values from the boundary

region of the fine grid, and thus not taken as reliable enough to replace part of the coarse flux determined exclusively from active zones of the coarse grid.

Equation (13) assumes that both the left and right faces of the coarse ROI need to be flux corrected. Should the fine grid in Figure 4d not cover coarse zones (I, J, K) and $(I+1, J, K)$, flux corrections need only be applied to the left side of the coarse ROI (I, J, K) . In this case, the terms $F_{1,S_2}^{N+1/2}(I+1, J, K)$ and $f_{1,s_2}^{n+\tau+1/2}(i+\nu, j+\beta, k+\eta)$ are omitted. Similarly, if the fine grid does not cover coarse zones $(I-1, J, K)$ and (I, J, K) , the terms $F_{1,S_2}^{N+1/2}(I, J, K)$ and $f_{1,s_2}^{n+\tau+1/2}(i, j+\beta, k+\eta)$ are omitted.

The fourth and final case is when both the flux and momentum component are perpendicular to the fine grid skin normal. Here, no flux corrections are required, as can be readily seen by inspection of Figure 4c.

For the magnetic field components, the operator in the evolution equation (equation 3) is the curl, and not the divergence of the hydrodynamical variables. Thus, \vec{B} is a surface conserved quantity (rather than volume-conserved) and, as such, we make adjustments to the so-called EMFs (defined below) instead of the fluxes; otherwise the procedure is the same.

Consider the coarse ROI of B_3 immediately to the right of a fine grid (Figure 4e). Following Balsara (2001), we have:

$$\begin{aligned} \tilde{B}_3^{N+1}(I, J, K) = & B_3^{N+1}(I, J, K) + \frac{1}{\Delta A_3(I, J, K)} \left[\mathcal{E}_2^{N+\frac{1}{2}}(I, J, K) \right. \\ & \left. - \sum_{\beta, \tau=0}^{\nu-1} \varepsilon_2^{n+\tau+\frac{1}{2}}(i, j+\beta, k) \right], \end{aligned} \quad (14)$$

where $\mathcal{E}_2^{N+1/2}(I, J, K) = E_2 \Delta x_2 \Delta t$ is the time-centred coarse 2-EMF⁴ (“electro-motive force”) located along the 2-edge, and where $\varepsilon_2^{n+\tau+1/2}(i, j+\beta, k)$ is the time-centred fine 2-EMF, located along the same 2-edge as the coarse 2-EMF. Here, the quantity in square brackets are the “EMF corrections”, analogous to the flux corrections for the hydrodynamical variables. In *ZEUS-3D*, the EMFs can be evaluated using various algorithms. In our case, we use the Consistent Method of Characteristics (CMoC) described by Clarke (1996).

Because the magnetic field is an area-conserved quantity, there is no situation where the coarse ROI of a magnetic field component is partially covered by fine ROIs as can happen for

⁴Similar to the hydrodynamical fluxes, we define the EMFs with the factor Δt embedded to simplify the accounting.

the volume-conserved, face-centred momentum. Thus, equation (14) and its permutations are sufficient to cover all EMF corrections for all field components in all directions. Note, for example, there are no corrections to be made in the 1-direction for the 1-field, again because of the surface conservative nature of the magnetic field.

Further, induction of magnetic field components penetrating the skin of a fine grid is affected by EMFs computed from quantities taken entirely from within the active portion of the grid. This is in contrast to a momentum component penetrating the skin, half of whose fluxes are computed from boundary values. Thus, we take the skin values of the EMFs and the magnetic components they induce to be just as reliable as those evaluated from within the grid, and it is appropriate to restrict the coarse magnetic field values on the skin of a fine grid with the overlying fine values (*e.g.*, equation 9).

Finally, flux correction equations for coarse zones adjacent to other sides of a fine grid can be obtained by a suitable permutation of indices and subscripts.

4. Prolongation

Prolongation is the process by which fine grid zones are filled using the best information available. This could either be by interpolating values from the underlying coarse grid in a way to ensure local conservation and local monotonicity, or taking them from adjacent or overlapping fine grids. As with restriction, prolongation can be divided into two types. First, when a fine grid is created or extended, the new fine zones must be filled. Second, at the beginning of a fine grid time step, fine boundary zones must be set. Both types require interpolation methods, some of which have been introduced here specifically for static grid refinement.

4.1. Spatial interpolation

In an effort to minimise the errors caused by waves travelling across boundaries between fine and coarse grids, we have introduced higher-order interpolation schemes (*e.g.*, piecewise parabolic interpolation, PPI, Colella & Woodward 1984) to the prolongation step. This improves the results for adaptive refinement, and has proven important for static refinement where strong waves are required to cross grid boundaries. By design, these interpolation schemes honour conservation laws and monotonicity.

In all cases and for all variables, we begin by estimating the interface values $Q_{L,R}(I)$ from cubic fits to the coarse grid data (Section 1 of Colella & Woodward 1984) without the

monotonisation or steepening steps. Now, to fit a parabolic interpolation function, $q_1^*(x)$, across the coarse ROI for zone I , we need three constraints. Two come from requiring that $q_1^*(x)$ passes through $Q_{L,R}(I)$, while the third comes from requiring that the zone-average value $Q(I, J, K)$ times the zone volume equal the volume integral of $q_1^*(x)$. For the original PPI scheme, this final constraint is written as:

$$Q(I, J, K)\Delta x_1(I) = \int_{\Delta x} q_1^*(x)dx.$$

For our purposes, we need to interpolate fine zone *averages* from the coarse ones and, as such, the conserved quantity is the *Riemann sum* and not the integral. Thus, in the 1-direction, our final constraint is:

$$Q(I, J, K)\Delta x_1(I) = \sum_{\alpha=0}^{\nu-1} q_1^*(i + \alpha, j, k) \delta x_1(i) \quad (15)$$

$$\Rightarrow Q(I, J, K) = \frac{1}{\nu} \sum_{\alpha=0}^{\nu-1} q_1^*(i + \alpha, j, k) \quad (16)$$

for constant $\Delta x, \delta x$. With this, our parabolic interpolation function is (*c.f.*, equation 1.4 in [Colella & Woodward 1984](#)):

$$q_1^*(i + \alpha) = Q_L(I) + \zeta (Q_R(I) - Q_L(I) + \mathcal{H}_1(1 - \zeta)), \quad (17)$$

where:

$$\begin{aligned} \zeta &= \frac{x_1(i + \alpha) - x_1(I)}{\Delta x_1(I)}; \\ \mathcal{H}_1 &= \frac{1}{f_1(\nu) - f_2(\nu)} \left(Q(I, J, K) - Q_L(I) - f_1(\nu)(Q_R(I) - Q_L(I)) \right); \\ f_1(\nu) &= \frac{1}{2\nu^2} \sum_{\xi=1}^{\nu} (2\xi - 1) = \frac{1}{2}; \\ f_2(\nu) &= \frac{1}{4\nu^3} \sum_{\xi=1}^{\nu} (2\xi - 1)^2 = \frac{4\nu^2 - 1}{12\nu^2}. \end{aligned} \quad (18)$$

With $q_1^*(i + \alpha)$ determined, we set the differences between the fine and coarse zones:

$$\delta q_1(i + \alpha) = q_1^*(i + \alpha) - Q(I, J, K). \quad (19)$$

For zone-centred quantities, we determine the fine-zone differences in the manner of equation (19) in each of the three directions, and set the interpolated fine zone averages to be:

$$q(i + \alpha, j + \beta, k + \eta) = Q(I, J, K) + \delta q_1(i + \alpha) + \delta q_2(j + \beta) + \delta q_3(k + \eta). \quad (20)$$

Given equation (16) and analogous expressions in the 2- and 3-directions, it is easy to show that this prescription is conservative; that is,

$$Q(I, J, K) = \frac{1}{\nu^3} \sum_{\alpha, \beta, \eta=0}^{\nu-1} q(i + \alpha, j + \beta, k + \eta), \quad (21)$$

for uniform grids ($\Delta V = \nu^3 \delta V$).

For face-centred quantities, we determine the fine-zone differences in the manner of equation (19) in each of the two orthogonal directions (*e.g.*, in the 2- and 3-directions for S_1), and set the interpolated fine zone averages along each coarse face to be:

$$s_1(i, j + \beta, k + \eta) = S_1(I, J, K) + \delta s_{1,2}(j + \beta) + \delta s_{1,3}(k + \eta). \quad (22)$$

Exactly the same procedure is used to interpolate B_1 across each coarse 1-face. As with the zone-centred quantities, it is easy to show that this procedure conserves momentum (magnetic) flux on the face:

$$S_1(I, J, K) = \frac{1}{\nu^2} \sum_{\beta, \eta=0}^{\nu-1} s_1(i, j + \beta, k + \eta). \quad (23)$$

The next step is to interpolate the face-centred quantities into the interior of the zone, and it is here where the procedure for momentum and magnetic components diverge.

For the volume-conserved momenta, we perform a linear interpolation between the fine momenta on opposing coarse faces. For example, between $s_1(i, j + \beta, k + \eta)$ and $s_1(i + \nu, j + \beta, k + \eta)$ we set:

$$s_1(i + \alpha, j + \beta, k + \eta) = (1 - \zeta) s_1(i, j + \beta, k + \eta) + \zeta s_1(i + \nu, j + \beta, k + \eta), \quad (24)$$

where $\alpha = 1, \dots, \nu - 1$ and $\zeta = \alpha \delta x / \Delta x$. This prescription guarantees that over a *zone-centred volume*, the prolongation of the 1-momentum is conservative. The fact that conservation is over the zone-centred volume and not specifically the 1-momentum ROI is required for equations (23) and (24) to be consistent with the restriction procedure of equation (11).

For the area-conserved magnetic field, a simple linear interpolation between opposing coarse faces does not preserve the solenoidal condition. Thus, we turn to a generalised, directionally unsplit version of the algorithm described by Li & Li (2004) in which $\nabla \cdot \vec{b} = 0$ so long as $\nabla \cdot \vec{B} = 0$ holds on the underlying coarse grid.

In this approach, we take the coarse and recently interpolated fine values of magnetic field (equation 22), and apply the solenoidal condition to determine “intermediate” coarse

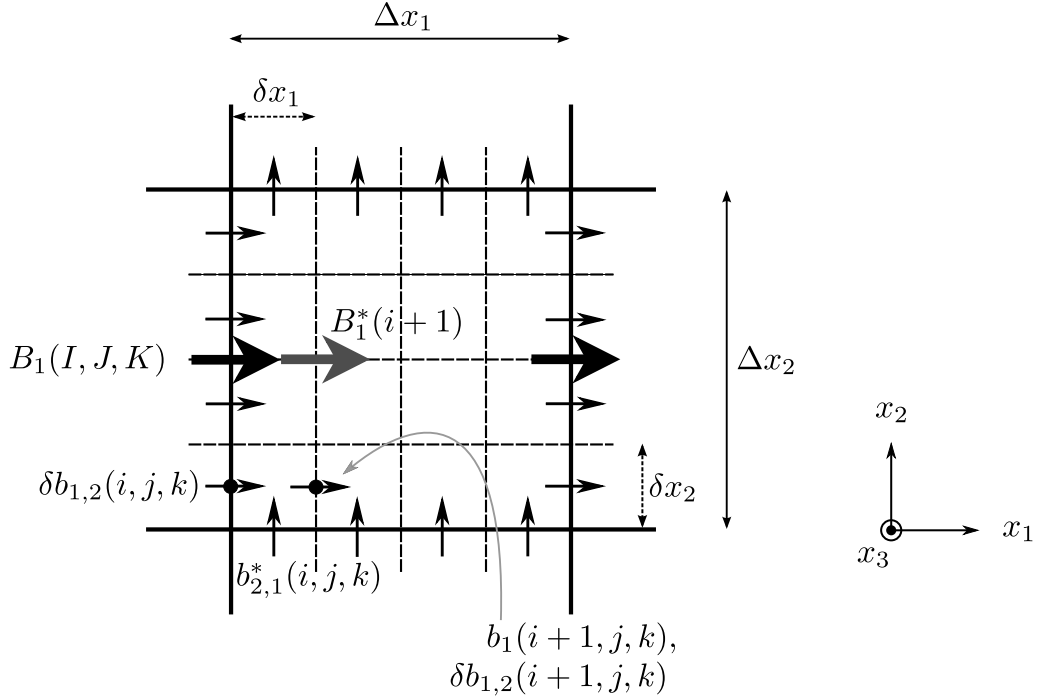


Fig. 5.— Schematic representation of the directionally unsplit **Li & Li** algorithm for calculating fine values of \vec{b} between coarse grid faces when $\nu = 4$.

field values (*e.g.*, B_1^* ; Figure 5), which are cospatial with fine zone faces. For example, the intermediate values for B_1 are given by:

$$B_1^*(i + \alpha + 1) = B_1^*(i + \alpha) - \delta x_1(i + \alpha) \left(\frac{b_{2,1}^*(i + \alpha, j + \nu, k) - b_{2,1}^*(i + \alpha, j, k)}{\Delta x_2(J)} + \frac{b_{3,1}^*(i + \alpha, j, k + \nu) - b_{3,1}^*(i + \alpha, j, k)}{\Delta x_3(K)} \right), \quad (25)$$

where:

$$\begin{aligned} b_{2,1}^*(i + \alpha, j, k) &= \frac{1}{\nu} \sum_{\eta=0}^{\nu-1} b_2(i + \alpha, j, k + \eta), \\ b_{3,1}^*(i + \alpha, j, k) &= \frac{1}{\nu} \sum_{\beta=0}^{\nu-1} b_3(i + \alpha, j + \beta, k); \\ B_1^*(i + 0) &= B_1(I, J, K), \end{aligned} \quad (26)$$

and $0 \leq \alpha \leq \nu - 2$. Given $B_1^*(i + \alpha + 1)$, and the differences between fine and coarse values at the coarse faces ($\delta b_{1,l}$, $l = 2, 3$; equation 19), we then calculate the fine magnetic field at $i + \alpha + 1$:

$$b_1(i + \alpha + 1, j + \beta, k + \eta) = B_1^*(i + \alpha + 1) + \delta b_{1,2}(i + \alpha + 1, j + \beta, k) + \delta b_{1,3}(i + \alpha + 1, j, k + \eta), \quad (27)$$

where:

$$\begin{aligned} \delta b_{1,2}(i + \alpha, j + \beta, k) &= (1 - \zeta) \delta b_{1,2}(i, j + \beta, k) + \zeta \delta b_{1,2}(i + \nu, j + \beta, k), \\ \delta b_{1,3}(i + \alpha, j, k + \eta) &= (1 - \zeta) \delta b_{1,3}(i, j, k + \eta) + \zeta \delta b_{1,3}(i + \nu, j, k + \eta), \end{aligned} \quad (28)$$

and $\zeta = \alpha \delta x / \Delta x$.

Proceeding incrementally from $i + 1$ to $i + \nu - 1$ provides all of the values of $b_1(i, j, k)$ between the coarse faces (I, J, K) and $(I + 1, J, K)$. Values for the other magnetic field components are given by a straightforward permutation of indices. In aggregate, these yield a third-order interpolation of fine magnetic field components that preserve the divergence of the underlying coarse magnetic field to machine round-off error.

Finally, a note on the use of vector potentials. [Igumenshchev & Narayan \(2002\)](#) have demonstrated that the vector potential, \vec{A} , may be used in the CT algorithm ([Evans & Hawley 1988](#)) instead of the magnetic field with results identical to machine round-off error. Thus, one may be tempted to adopt this approach so that prolongation methods simpler than the [Li & Li](#) algorithm may be used on \vec{A} as a way to guarantee that the fine grid satisfies the solenoidal condition. Reasons for *not* taking this approach are given in Appendix B.

4.2. Temporal interpolation

Fine grids require prolonged boundary values at the beginning of each fine time step, and thus we also need to perform temporal interpolations on the coarse values. Following [BC89](#), we perform a linear interpolation on the coarse hydrodynamic variables in time, and then spatially interpolate them as described above to obtain the necessary boundary information for each fine time step.

For the magnetic field, the fine components penetrating the skins of the fine grid are retained; only those values completely within the boundary region are prolonged from the coarse grid. For this, we require coarse grid EMFs on the fine grid skin and within the boundary region that are time centred on the current fine time step.

The coarse EMFs coincident with the fine grid skin are replaced with the spatial *and* temporal sum of the overlying fine EMFs, where the sum over time is necessary since the

time step is embedded in our definition (footnote 4). Thus and for example, for \mathcal{E}_2 we set:

$$\mathcal{E}_2^{N+\psi}(I, J, K) = \sum_{\tau=0}^{\tau'} \sum_{\beta=0}^{\nu-1} \varepsilon_2^{n+\tau+\frac{1}{2}}(i, j + \beta, k), \quad (29)$$

where $\psi = (2\tau' + 1)/2\nu$, and where $0 \leq \tau' \leq \nu - 2$ designates the last completed fine time step. This yields values on the fine grid skin which are time-centred between the beginnings of the coarse time step and the current fine time step, and take into account the superior quality of the fine grid data.

The coarse EMFs residing entirely within the fine grid boundary region are retained. However, these are time-centred for the coarse grid and, as such, have embedded in them a factor Δt . To render them coeval with fine time step τ' and thus the coarse skin EMFs computed in equation (29), we must multiply them by $(2\tau' + 1)/\nu = 2\psi$ to correct the embedded time step factor.

With coarse EMFs on the fine skin and inside the fine boundary region properly time-centred, the coarse magnetic field components are updated to the beginning of the current fine time step using equation (3), and it is these values that are used in the prolongation methods described in Section 4.1 to create the fine grid magnetic field boundary values.

One might ask why a linear temporal interpolation of the divergence-free coarse magnetic field components isn't enough to preserve the solenoidal condition within the boundary region. Indeed, such an approach does guarantee $\nabla \cdot \vec{B} = 0$ *within the region interpolated*, but not in the layer between the interpolated region and its immediate non-interpolated neighbours. The method outlined above based on the coarse EMFs allows interpolations to be performed locally and only where they are needed, while still preserving the solenoidal condition globally to machine round-off error.

4.3. Monotonicity

We have found it necessary to maintain a certain monotonicity in our prolongations. Failure to do so can lead to negative pressures (even when solving the internal energy equation) and violations of the CFL condition. For example, if the current time step is governed by the sound speed, and the prolongation process leads to an interpolated density less than the surrounding zones, the sound speed in a fine zone could be greater than that which was used in determining the CFL time step, possibly leading to numerical oscillations and loss of stability.

Sequentially stringing together two (or three) 1-D PPIs as we do for prolongation of

zone-centred quantities (*e.g.*, equation 20) can lead to non-monotonic behaviour even if each 1-D interpolation is separately monotonic. If a PPI-determined value $q(i + \alpha, j + \beta, k + \eta)$ is found to lie outside the range set by neighbouring coarse zones:

$$\left. \begin{aligned} Q_{\min} &= \min (Q(I + \Gamma, J + \Lambda, K + \Upsilon)); \\ Q_{\max} &= \max (Q(I + \Gamma, J + \Lambda, K + \Upsilon)), \end{aligned} \right\} -1 \leq \Gamma, \Lambda, \Upsilon \leq +1,$$

then we “fall back” to piece-wise linear interpolations (PLI; van Leer 1977, Appendix A). In rare cases where the PLI-determined value is also non-monotonic (possible only in 3-D), we revert to piece-wise constant interpolations (PCI, *a.k.a.* “direct injection” or “donor cell”):

$$q(i + \alpha, j + \beta, k + \eta) = Q(I, J, K).$$

Where PPI yields non-monotonic results, we note that transmission of strong waves across changes in resolution (*e.g.*, static grids) is significantly improved if one first tries PLI rather than falling back directly to PCI or limiting the interpolated value to lie between Q_{\min} and Q_{\max} .

5. Boundary conditions

Boundary conditions are applied directly to the hydrodynamical variables (ρ, e, \vec{v}) and indirectly to the magnetic field via the EMFs. Attempting to apply boundary conditions directly to \vec{B} often generates monopoles in the boundary regions, which can have significant effects on the dynamics in the active grid, as illustrated in Appendix C.

In addition to the usual boundary conditions applied by the *ZEUS* module during each MHD cycle, the AMR module must also set boundary conditions on two occasions. After restriction and before prolongation, boundary values are set on the coarse grid. Then, after prolongation and before the *ZEUS* module is called for the next MHD cycle, boundary values for the fine grid are set using the results of the prolongation step. These additional applications of boundary conditions are necessary to maintain the imposed physical boundaries after restriction and prolongation have altered some of the values on the active grid, and to reconcile boundaries of grids that may be contained within or adjacent to other grids (henceforth referred to as *adjacent boundaries*).

Because of their nature, physical boundaries (inflow, outflow, reflecting; those traditionally applied by single-grid MHD codes), and adjacent boundaries must be treated differently, and each is discussed in turn.

5.1. Physical Boundaries

Generally, a fine grid is completely embedded within a coarse grid. The single exception is when both grids share a physical boundary and two items of note must be borne in mind when adapting the physical boundary condition routines in *ZEUS-3D* to *AZEuS*.

First, since each grid has only two boundary zones, only part of the coarse boundary region is covered by fine boundary zones. Thus, coarse boundary zones cannot be included in the restriction step; a particular concern when setting magnetic boundary conditions. In *AZEuS*, we extend the EMF correction scheme of Section 3.2 by retaining *three* layers of transverse EMF corrections, and *two* layers of longitudinal EMF corrections⁵, including the skin layer plus two additional layers interior and adjacent to the skin. Immediately before the restriction step, physical boundary conditions are applied to the EMF corrections, which are then used to update the boundary values of the magnetic field components in the coarse grid according to equation (14). Done in this fashion, there is no risk of introducing monopoles to the coarse boundary zones during the restriction step when two or more grids overlap a physical boundary.

Second, not handled properly, inflow boundary conditions can introduce unexpected violations of conservation laws which can cause unwanted discontinuities in the boundary. In particular, if a boundary variable is to be set according to an analytical function of the coordinates, that variable should be set to the *zone average* of that function, and not simply to the function value at the location of the variable. While this is good advice for a single grid application, it is critical for AMR.

For example, suppose the density profile:

$$\rho(r) = \frac{1}{r^{3/2}}, \quad (30)$$

is to be maintained in cylindrical coordinates along the $z = 0$ boundary. Let $\rho(J)$ be the density in the coarse zone of dimension $(\Delta z, \Delta r)$ centred at $r(J) = r$, and let $\rho(j)$ and $\rho(j+1)$ be the densities in the fine zones of dimension $(\delta z, \delta r)$ centred at $r(j)$ and $r(j+1)$ (Figure 6). For a refinement ratio of 2, $\Delta r = 2\delta r$, $\Delta z = 2\delta z$, $r(j) = r - \frac{1}{2}\delta r$, and $r(j+1) = r + \frac{1}{2}\delta r$.

If we naïvely set $\rho(J)$ using equation (30), the mass of zone J (with volume $\Delta V = r\Delta r\Delta z$) is:

$$m(J) = \rho(J)\Delta V(J) = \frac{\Delta r\Delta z}{r^{1/2}}.$$

⁵*e.g.*, the transverse (longitudinal) EMFs for the 1-boundary are ε_2 and ε_3 (ε_1).

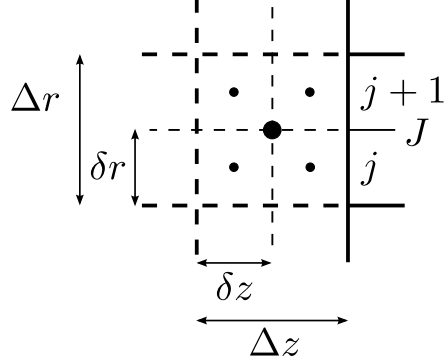


Fig. 6.— A single coarse zone in the $z = 0$ boundary with zone centre at $r(J) = r$ and with refinement ratio of $\nu = 2$.

Similarly, for the fine zones,

$$m(j) = \frac{\delta r \delta z}{(r - \frac{1}{2}\delta r)^{1/2}}, \quad m(j+1) = \frac{\delta r \delta z}{(r + \frac{1}{2}\delta r)^{1/2}}.$$

Adding the masses in the four fine zones contained by the coarse zone, we find:

$$2(m(j) + m(j+1)) = m(J) \left(1 + \frac{3}{32} \left(\frac{\delta r}{r} \right)^2 + \dots \right) > m(J),$$

and the physical boundary conditions violate mass conservation.

Instead, $\rho(J)$ should be set to the *zone average*, determined by:

$$\rho(J) = \frac{m(J)}{\Delta V(J)}$$

where the mass function, $m(J)$, is given by:

$$m(J) = \int_{\Delta V(J)} \rho(r) dV, \tag{31}$$

where $\rho(r)$ is the given density profile (*e.g.*, equation 30). Similar expressions apply for $m(j)$, and $m(j+1)$. Done in this fashion, it is easy to show that

$$\rho(J) \Delta V(J) = 2(\rho(j+1) \delta V(j+1) + \rho(j) \delta V(j)),$$

and both grids agree on the mass contained within coarse boundary zone J to within machine round-off error. With a little bit of algebra, this can be confirmed analytically for the specific case of equation (30). For more complicated profiles, a numerical integrator can be employed to perform the necessary integrations in equation (31).

5.2. Adjacent Boundaries

Unlike Godunov methods which typically require a single application of boundary conditions at the end of each MHD cycle, the operator-split nature of *ZEUS* requires boundary conditions to be set several times. For physical boundaries, this is not an issue; physical boundary conditions may be set whenever needed based on data from a single grid. However, adjacent boundaries pose a unique challenge in *AZEuS* since the *ZEUS* module is aware only of the grid being updated, and other grids cannot be accessed in the middle of an MHD cycle to update these boundaries.

We have introduced into *AZEuS* the concept of “self-computing” boundary conditions for adjacent boundaries. In this approach, boundary zones are set at the beginning of each time step using the best information available (either from an adjacent grid of the same resolution, or from the prolongation of an underlying coarse grid possibly interpolated in time), and then the full set of operator-split MHD equations are applied to both the boundary and active zones; no adjacent boundary zones are reset to any pre-determined quantity inside a single MHD time step. Further, self-computed boundary zones are included in the calculation of the CFL time step, which we find critical (for an operator-split code such as *AZEuS*) in minimising transmission errors when waves of any significant amplitude cross adjacent boundaries.

Of course, assumptions about missing data beyond the outermost edge of the grid must be made, and “pollution” from these missing data necessarily propagates inward. For example, consider the left 1-boundary where $v_1(i = 1)$ and $v_1(i = 2)$ represent the boundary values of v_1 , $v_1(i = 3)$ the skin value (treated as a boundary value), and $v_1(i = 4)$ the first active value. The pressure gradient at $v_1(1)$ is proportional to $p(1) - p(0)$, yet $p(0)$ is completely unknown [$p(1)$ and $p(2)$ are the only boundary values available]. The best we can do for a missing datum such as this is to extrapolate assuming a zero-gradient, in which case $p(0) = p(1)$ and no pressure gradient is applied to $v_1(1)$. In this manner, $v_1(1)$ is polluted by the missing datum at the very beginning of the source step.

Additional steps inside a single MHD cycle include the application of artificial viscosity, adiabatic expansion/compression for the internal energy equation, the transport steps, and the induction step. All but the latter contribute to propagating pollution from missing data toward the active portion of the grid. Ideally, one would carry enough boundary zones to prevent pollution from reaching the active grid within a single MHD step, so that the AMR module (with knowledge of all grids) can reset all boundary values to new and unpolluted values before their effects ever reach the active zones. However, for [van Leer \(1977\)](#) interpolation in the transport step and within a single MHD cycle, pollution from missing data reaches the skin and first active zone-centre on the grid, as well as the first

active face and second active zone-centre if the local velocity points away from the boundary (true regardless of which energy equation is used). Thus, to completely prevent pollution from reaching the active grid, we would have to double the number of boundary zones from two to four.

We have investigated this effect thoroughly and have found no test problem that demonstrates anything but the slightest quantitative effect from pollution by missing data at adjacent boundaries between fine and coarse grids. For adjacent boundaries between grids of like resolution, we have elected to force these grids to overlap (for reasons explained in Section 6) by an amount sufficient to eliminate the problem of missing data pollution altogether.

6. Grid creation and adaptation

Creation or modification of adaptive grids in AZEuS proceeds in a manner similar to Bell *et al.* (1994) including the suggestions of Berger & Rigoutsos (1991), with two important differences. First, we have had to modify the proper nesting criterion of BC89 by increasing from one to three the number of zones at level $l - 1$ separating an active zone at level l from level $l - 2$. This is because prolongation of boundary values for level l from level $l - 1$ requires one zone from level $l - 1$ beyond the edge of a grid at level l , plus two additional zones at level $l - 1$ on either side to satisfy the five-zone molecule needed by PPI.

Second, BC89 allow grids of the same resolution to abut without overlapping, whereas we have found it advantageous for at least two reasons to extend abutting grids so that they overlap by a minimum of one coarse zone (Figure 7). For one, since the momenta penetrating a grid skin are no more reliable than boundary values, two abutting grids would, in general, disagree on the values of the momentum penetrating their common skin. This turns out to pose an intolerable ambiguity in the solution. Second, the problem of pollution propagating from missing data onto the active grid (Section 5.2) is completely averted by overlapping two abutting grids by one coarse zone.

By forcing abutting grids to overlap, there is always a clear choice of which value to use at a given location. Where the boundary and skin values of one grid overlap the active zone values of another, the active zone values prevail, and are used by the other grid for its boundary values.

Grids which are abutting at the end of the grid generation and modification process, but before they are prolonged, are made to overlap by one coarse zone. Numerous situations arise in which more than one grid may overlap at the same location (*e.g.*, the test problems in sections 7.2.1 and 7.2.2), and this can result in overlaps which are greater than one coarse

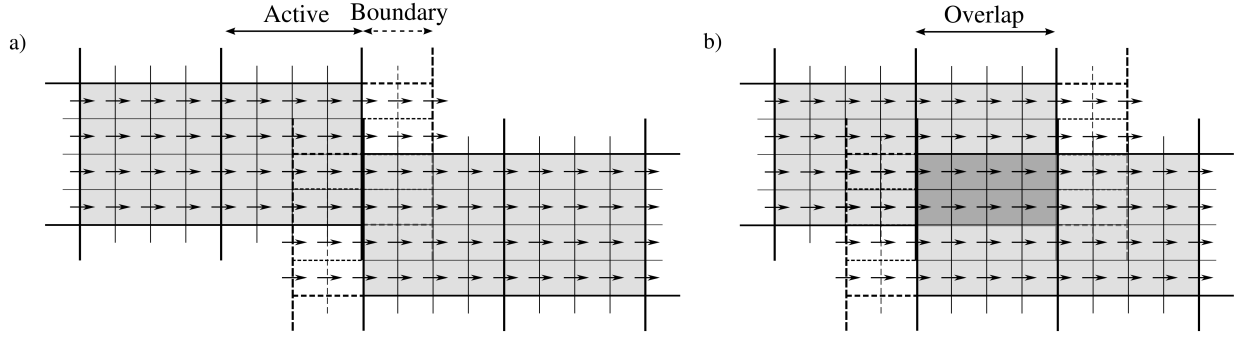


Fig. 7.— Two grids that originally abut (panel a) are made to overlap by at least one coarse zone (panel b).

zone. Further, the possibility of a complex distribution of AMR grids opens up a whole host of pedagogical cases which one must consider, particularly when looking to maintain the solenoidal condition. During the development of AZEuS, we have carefully examined each case involving multiple levels with numerous grids overlapping each other, to ensure that where grids overlap, they all agree on the flow variables and all conservation laws are preserved to machine round-off error.

In this regard, the EMFs turn out to be a very sensitive discriminator. Any mismatches between overlapping grids in the conserved variables at the beginning of a time step will first generate mismatched velocities, followed by disagreements in co-spatial EMFs before ν time steps have passed. These can be detected, for example, as monopoles arising in either the boundary or active zones in either or both of the overlapping grids. Even if such monopoles are restricted entirely to the boundary zones, their unphysical forces can affect neighbouring active zones whose effects propagate rapidly throughout the grid (*e.g.*, Appendix C). Preserving agreement between overlapping zones including boundary zones to machine round-off is, therefore, of paramount importance.

7. Numerical Tests

We have verified AZEuS against a number of standard test problems, some of which are presented in this section. Further results will be posted to <http://www.ica.smu.ca/azeus>, as they become available. Additional and similar test problems for ZEUS-3D (without AMR) are found in Clarke (2010) and on-line at <http://www.ica.smu.ca/zeus3d>.

7.1. 1-D shock tubes

While most AMR applications rely exclusively on dynamic grids, certain applications, particularly those which exhibit some degree of self-similarity, can benefit enormously from the use of a base of nested, static grids (*e.g.*, Ramsey & Clarke 2011). As such solutions evolve, waves of all types—including strong shocks—must pass sequentially from one grid to another, and it is here where our higher order prolongation algorithms for adjacent boundaries are critical.

AZEuS has been tested with all 1-D shock tube problems from Ryu & Jones (1995; RJ95) for both static and dynamic grids, and we show the results from two to highlight these features of the code. Both tests use the total energy equation with $\gamma = 5/3$, $\mathcal{C} = 0.75$ (Courant number), and artificial viscosity parameters `qcon` = 1.0 and `qlin` = 0.2.

For static grids, Figure 8 shows two solutions for ρ , e_T , v_2 , and B_2 from problem (4a) of RJ95 over a domain of $x_1 \in [-0.5, 2.5]$ (three times larger than RJ95) for a time $t = 0.45$ (three times longer). The initial discontinuity is placed at $x_1 = 0.5$, with the left and right states given in the figure caption. The left panels show the domain resolved with 1200 zones and no AMR, while the central panels show the AMR solution with a base resolution of 600 zones and fine static grids with a refinement ratio of 2 placed at $x_1 \in [-0.2, 0.1]$ and $x_1 \in [0.7, 1.245]$ (grey). These locations allow all the physical features, with the exception of the sluggish slow rarefaction, to suffer at least one change in resolution. The right panels show the percent differences between the two solutions.

Discounting all zones trapped within a discontinuity (which, even without AMR, are already in error by as much as 100% since discontinuities are supposed to be infinitely sharp), the maximum error one can attribute to the use of static grids in any of the variables is less than 1%. As further evidence of the ability of AZEuS to pass waves of all types across adjacent boundaries, Ramsey & Clarke (2011) find almost no sign of reflection, refraction, or distortion of any type of wave across any of the static grid boundaries in their 2-D axisymmetric simulations of protostellar jets in cylindrical coordinates.

For dynamic grids, Figure 9 illustrates the results of test problem (2a) of RJ95. We employ four levels of refinement (sequentially darker shades of grey) above the base grid with a resolution of 40 zones over the domain $x_1 \in [0.0, 1.0]$. The initial discontinuity is located at $x_1 = 0.5$, with the left and right states given in the figure caption. To track the developing features, refinements are based on three criteria (Khokhlov, 1998): contact discontinuities (CD), shocks, and gradients in certain variables, each requiring a threshold value to be set above which a zone is flagged for refinement. Here, we set parameters `tolcd` = `tolshk` = 0.1 for CDs and shocks, respectively, and check v_2 for gradients above `tolgrad` = 0.1. For

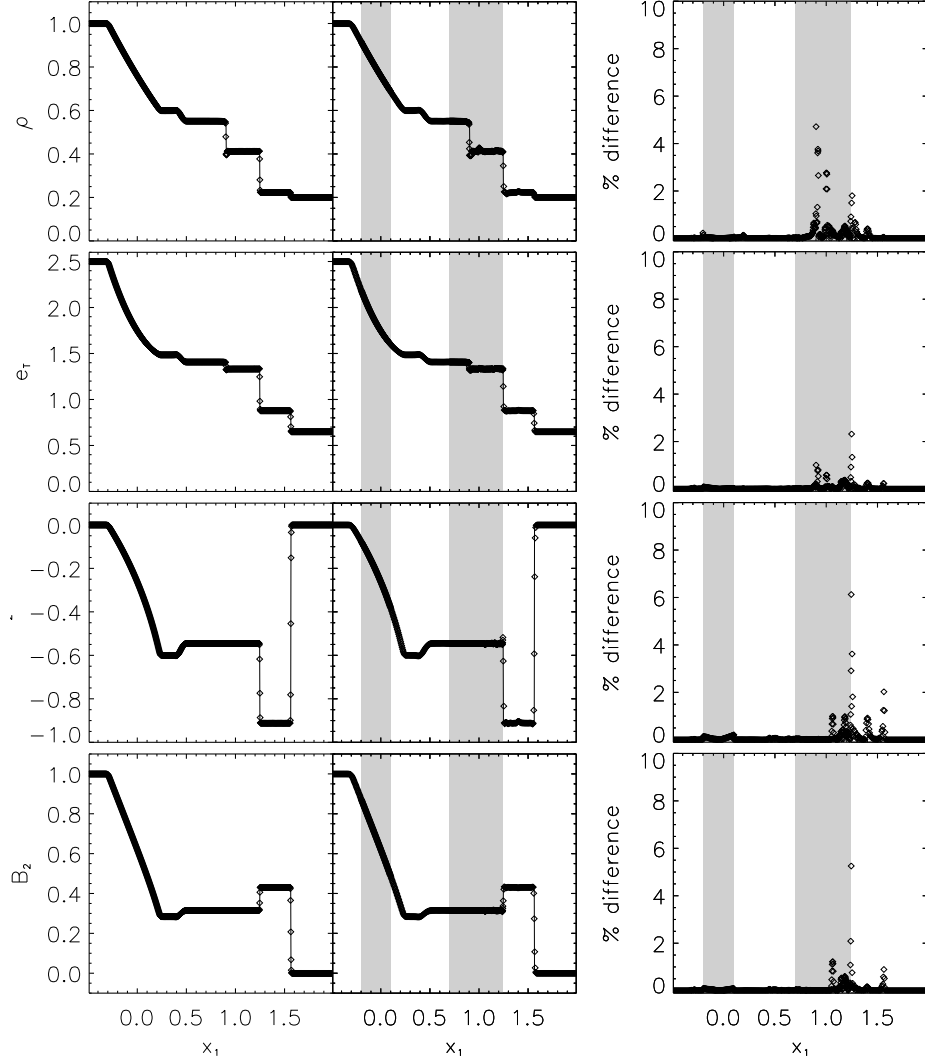


Fig. 8.— Static grid solution to problem (4a) of [RJ95](#) at time $t = 0.45$. The initial left and right states are $(\rho, v_1, v_2, v_3, B_2, B_3, p) = (1, 0, 0, 0, 1, 0, 1)$ and $(0.2, 0, 0, 0, 0, 0, 0.1)$, with $B_1 = 1$. From left to right, the physical features are: (1) fast rarefaction, (2) slow rarefaction, (3) contact discontinuity, (4) slow shock, and (5) “switch-on” shock. *Left panels*: uniform “fine” grid solution; *middle panels*: same solution with two fine, static grids (gray) overlying the coarse grid; *right panel*: percent difference between the uniform and static grid solutions. Solid lines are the analytical solutions from the Riemann solver described in [RJ95](#).

this problem, the gradient detector is needed to refine both the discontinuity of the initial conditions as well as rotational discontinuities ($x_1 \simeq 0.53$ and $x_1 \simeq 0.71$ in Figure 9), neither

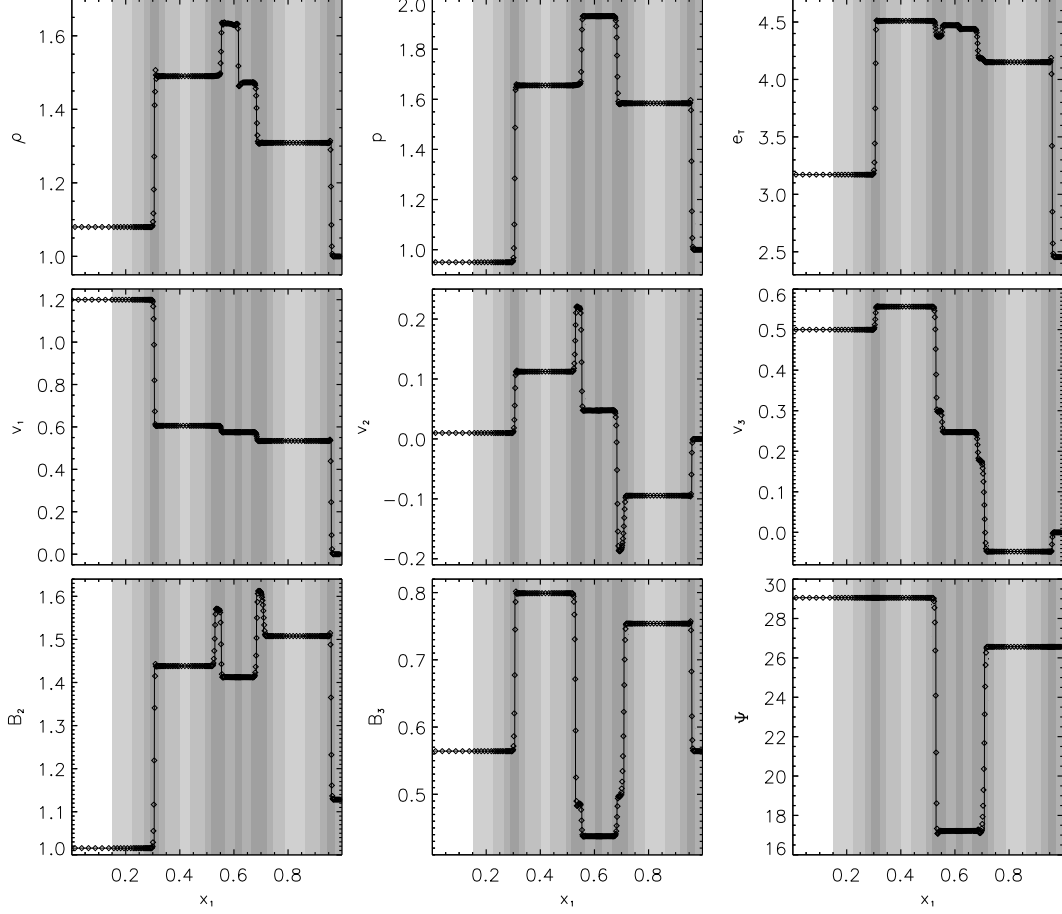


Fig. 9.— AZEuS solution to problem (2a) of [RJ95](#) at $t = 0.20$. The initial left and right states are $(\rho, v_1, v_2, v_3, B_2, B_3, p) = (1.08, 1.2, 0.01, 0.5, 3.6/\sqrt{4\pi}, 2/\sqrt{4\pi}, 0.95)$ and $(1, 0, 0, 0, 4/\sqrt{4\pi}, 2/\sqrt{4\pi}, 1)$, with $B_1 = 2/\sqrt{4\pi}$. The physical features, from left to right, are: (1) fast shock, (2) rotational discontinuity, (3) slow shock, (4) contact discontinuity, (5) slow shock, (6) rotational discontinuity, and (7) fast shock. Shaded regions indicate the location of finer grids, with the level of shading indicating the level of refinement. $\Psi = \tan^{-1}(B_3/B_2)$ is the angle between the transverse field components. The solid lines are the analytical solution from the Riemann solver described in [RJ95](#).

of which are detectable as a CD or shock.

Additional parameters used to obtain this solution include `kcheck` = 5 (the number of cycles between successive grid modifications), `geffcy` = 0.95 (the minimum allowed grid efficiency, defined as the ratio of zones flagged for refinement to the number of zones actually

parameter	description
\mathcal{C}	the Courant number
<code>qcon</code>	quadratic artificial viscosity parameter ^a
<code>qlin</code>	linear artificial viscosity parameter ^a
<code>tolcd</code>	threshold value for the CD detector (range: 0 – 1)
<code>tolshk</code>	threshold value for the shock detector (range: 0 – 1)
<code>tolgrad</code>	threshold value(s) for the gradient detector(s) (range: 0 – 1)
<code>kcheck</code>	number of cycles between successive grid modifications
<code>geffcy</code>	the minimum allowed fractional grid efficiency for creation of new grids (range: 0 – 1)
<code>ibuff</code>	the number of buffer zones added around zones flagged for refinement

Table 1: Summary and description of the significant user-set parameters in AZEuS.

^asee [Clarke 2010](#) for a more detailed description of the artificial viscosity parameters.

present in a new grid), and `ibuff` = 2 (the number of buffer zones added around flagged zones). These parameters, in addition to the non-AMR parameters, are summarised in [Table 1](#).

7.2. 2-D tests

7.2.1. MHD Blast

The MHD blast problem of [Londrillo & Del Zanna \(2000\)](#) and [Gardiner & Stone \(2005\)](#) has proven to be a very valuable test of our AMR algorithms and for rooting out problems in the code. It is also a good test of directional biases and the ability of a code to handle the evolution of strong MHD waves. We initialise the problem in the same manner as [Clarke \(2010\)](#) with domain $x \in [-0.5, 0.5]$, $y \in [-0.5, 0.5]$ and $(\rho, \vec{v}, B_1, B_2, B_3) = (1, 0, 5\sqrt{2}, 5\sqrt{2}, 0)$ everywhere. Within radius $r = 0.125$ of the origin, we set the gas pressure to $p = 100$, and $p = 1$ elsewhere.

Figure [10](#) presents our results for the 2-D blast problem at $t = 0.02$ for a base grid of 200^2 zones, and one additional level of refinement with ratio $\nu = 2$. For this test, we have purposely chosen a high grid efficiency of `geffcy` = 0.92 to strenuously test the ability of AZEuS to handle a large number of grids in a complicated pattern and refine on features

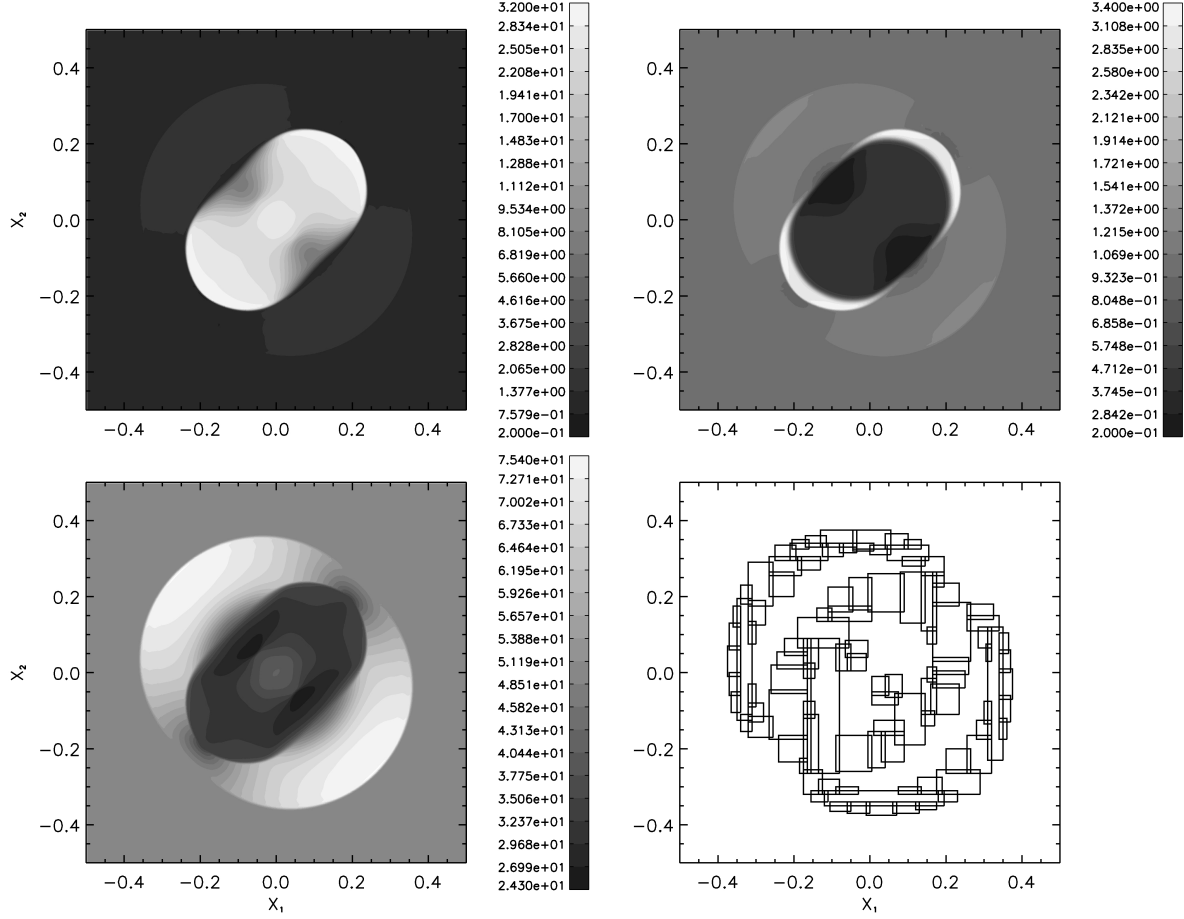


Fig. 10.— AZEuS solution for the 2-D MHD blast problem at $t = 0.02$ using 2 levels of refinement. *Top left*: gas pressure; *bottom left*: magnetic pressure ($p_B = |B|^2/2$); *top right*: gas density; and *bottom right*: distribution of AMR grids at $t = 0.02$.

which are not preferentially aligned with a coordinate axis. Typically, a lower value for the grid efficiency is used (*e.g.*, `geffcy` $\simeq 0.7$) to try and balance minimising the number of refined zones with the overhead associated with managing an increasing number of small grids.

Evidently, AZEuS is able to follow the blast wave closely, regardless of its orientation. Table 2 compares the extrema of the plotted variables between the AMR calculation and uniform grid solutions with 200^2 and 400^2 zones. With the exception of the pressure, which exhibits differences between the uniform 400^2 and AMR solutions of $\lesssim 0.5\%$, the uniform

	AMR		Uniform, 200 ²		Uniform, 400 ²	
	Min	Max	Min	Max	Min	Max
ρ	0.199	3.36	0.200	3.22	0.189	3.40
p	0.712	32.3	0.771	32.0	0.714	32.1
p_B	24.3	75.7	24.9	76.0	23.6	75.6

Table 2: Extrema for density, ρ , gas pressure, p , and magnetic pressure, p_B , in AMR and uniform grid solutions of the 2-D MHD blast problem at $t = 0.02$.

grid results bracket the AMR solution.

For this test, we set `tolshk` = `tolcd` = 0.2, and apply `tolgrad` = 0.2 to e_T . While the gradient detector is useful in refining the initial pressure jump, most ($\sim 99.9\%$) of the zones flagged for refinement soon thereafter are detected by the CD and shock detectors. Additional parameter settings include `kcheck` = 10, `ibuff` = 3, $\gamma = 5/3$, $\mathcal{C} = 0.5$, `qcon` = 1.0, and `qlin` = 0.1. All boundaries are set to outflow conditions.

7.2.2. Orszag-Tang MHD vortex

The 2-D vortex problem of [Orszag & Tang \(1979\)](#) has become a standard test for astrophysical MHD codes, and as it has not previously been performed using our version of *ZEUS-3D*, we present both non-AMR and AMR results here. It is important to note that the same code was used to produce both sets of results: *AZEuS* is designed to be modular, and by deselecting the ‘AMR’ option at the precompilation step, the code reverts to *ZEUS-3D*.

For this test, we follow [Stone *et al.* \(2008\)](#) by initialising a periodic, Cartesian box of size $x, y \in [0, 1]$ with initially constant pressure and density, $P = 5/12\pi$ and $\rho = \gamma P = 25/36\pi$, for $\gamma = 5/3$. The velocity is initialised to $(v_x, v_y, v_z) = (-\sin(2\pi y), \sin(2\pi x), 0)$, and the magnetic field is set through the vector potential $A_z = (B_0/4\pi) \cos(4\pi x) + (B_0/2\pi) \cos(2\pi y)$, where $B_0 = 1/\sqrt{4\pi}$.

The results for uniform grids with 256^2 and 512^2 zones at $t = 1/2$, as well as the AMR results for a base grid of 128^2 zones and 2 levels of refinement (effective resolution of 512^2 zones), are presented in Figure 11. The bottom right panel shows the distribution of AMR grids at $t = 1/2$. For clarity, we have only plotted the grids at level 3 (*i.e.*, 2 levels higher resolution than the base grid). Even then, the filling factor of level $l = 2$ grids at this time is $\gtrsim 95\%$. Examining the first three panels of Figure 11 closely, features are noticeably sharper in the 512^2 solution relative to the 256^2 results, while the AMR and 512^2 solutions

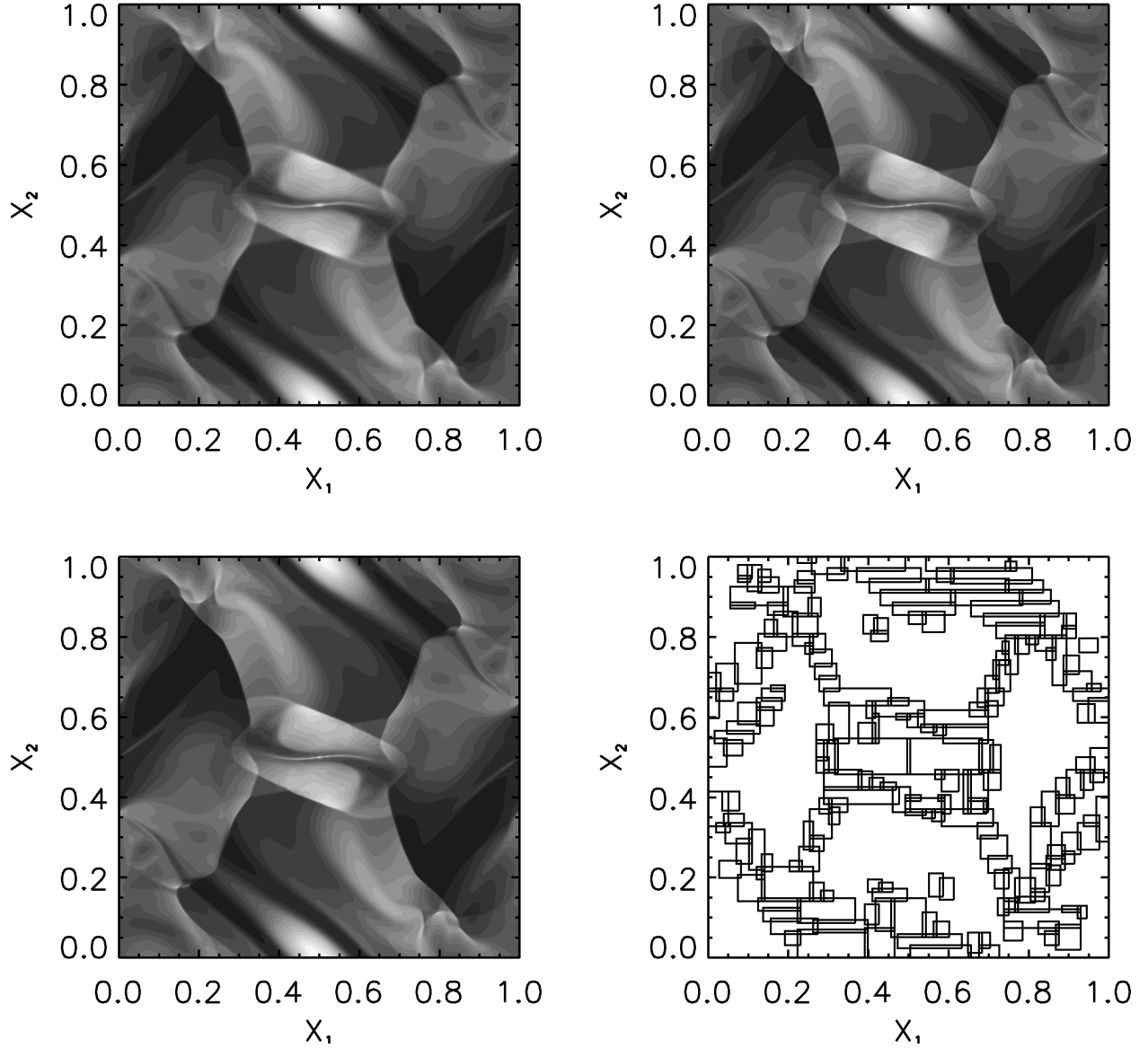


Fig. 11.— Uniform and adaptive grid solutions for the Orszag-Tang MHD vortex at $t = 1/2$. Plotted are 20 evenly spaced contours of the gas pressure with range $[0.03, 0.50]$. *Top left*: uniform grid solution with 256^2 zones; *top right*: uniform grid solution with 512^2 zones; *bottom left*: AMR solution with a base grid resolution of 128^2 and 2 levels of refinement; and *bottom right*: distribution of grids at level 3 in the AMR solution.

are indistinguishable.

Figure 12 shows slices of the gas pressure as a function of x at $t = 1/2$ and $y = 0.4277$,

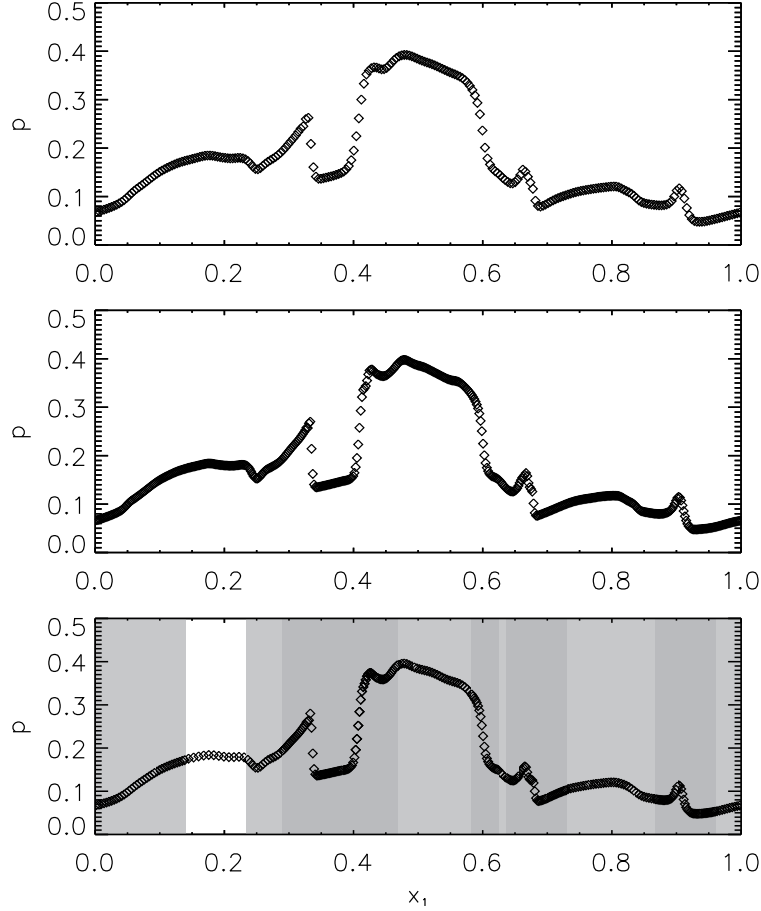


Fig. 12.— 1-D slices of the gas pressure at $t = 1/2$ and $y = 0.4277$ in the Orszag-Tang MHD vortex problem. From top to bottom, uniform 256^2 grid, uniform 512^2 grid, AMR solution with 2 levels of refinement and an effective resolution of 512^2 zones.

which once again demonstrate that the AMR and 512^2 uniform grid solutions are virtually identical. Quantitatively, these solutions compare favourably with those from higher-order codes such as *ATHENA* (Stone *et al.*, 2008), with the discontinuities in the AZEuS solutions being slightly broader.

For this problem, we set $\mathcal{C} = 0.5$, $q_{\text{con}} = 1.0$, and $q_{\text{lin}} = 0.1$ for both uniform and adaptive grids. For the AMR results, $k_{\text{check}} = 10$, $g_{\text{effcy}} = 0.9$, $i_{\text{buff}} = 2$, and $\text{tolshk} = 0.2$. Neither the CD nor gradient detector were engaged.

7.2.3. Magnetised accretion torus

This test is based on the simulations of [Hawley \(2000\)](#) and [Mignone *et al.* \(2007\)](#) for a magnetised, constant angular momentum torus in axisymmetric spherical (r, ϑ) coordinates, and highlights the use of curvilinear coordinates in AZEuS.

The torus structure is described by the equilibrium condition:

$$\frac{\gamma p}{(\gamma - 1) \rho} = C - \phi - \frac{1}{2} \frac{l_{\text{Kep}}^2}{r^2 \sin^2 \vartheta}, \quad (32)$$

where C is a constant of integration, $\phi = -1/(r - 1)$ is the pseudo-Newtonian gravitational potential, and l_{Kep} is the Keplerian angular momentum at the pressure maximum. The pressure p is initially related to the density via the polytropic relation, $p = \kappa \rho^\gamma$, with $\gamma = 5/3$. By specifying the location of the pressure maximum ($r_{\text{max}} = 4.7$) and the inner edge of the torus ($r_{\text{min}} = 3$), we can determine the value of C and the (hydrodynamic) structure of the torus.

A poloidal magnetic field is initialised in the torus from the φ -component of the vector potential:

$$A_\varphi = \frac{B_0}{\rho_m} \min(\rho(r, \vartheta) - \rho_c, 0),$$

where $B_0^2 = 2\kappa\rho_m^\gamma/\beta_m$, β_m and ρ_m are, respectively, the plasma-beta and density at r_{max} , κ is determined by equation (32) evaluated at r_{max} , and $\rho_c = \rho_m/2$ determines the surface of the last vector equipotential. For the simulations presented here, $\beta_m = 350$ and $\rho_m = 10$. The toroidal velocity (v_φ) in the torus is initialised to the local Keplerian speed, while the poloidal velocity is set to zero everywhere.

Outside the torus, the magnetic field is zero and we initialise a hydrostatic atmosphere with density and temperature contrasts of $\rho_{\text{atm}}/\rho_m = 10^{-4}$ and $T_{\text{atm}}/T_m = 100$, respectively, where T_m is the temperature at r_{max} .

The domain of the grid is $r \in [1.5, 20]$ and $\vartheta \in [0, \pi/2]$. We impose reflecting boundary conditions suitable for a rotation axis at $\vartheta = 0$, reflecting/conducting boundary conditions at the equatorial plane ($\vartheta = \pi/2$), and outflow boundary conditions at $r = 20$. At $r = r_{\text{in}} = 1.5$, we impose “sink” boundary conditions in an attempt to absorb any material reaching the inner boundary. This involves maintaining the density and pressure at their initial values, and setting $v_r = v_\vartheta = v_\varphi = 0$ within the boundary. On the inner skin ($r = r_{\text{in}}$), v_r is set to the minimum of zero and a linear extrapolation from the grid. Finally, outflow boundary conditions are applied to the EMFs:

$$\mathcal{E}_1(r < r_{\text{in}}) = \mathcal{E}_1(2r_{\text{in}} - r);$$

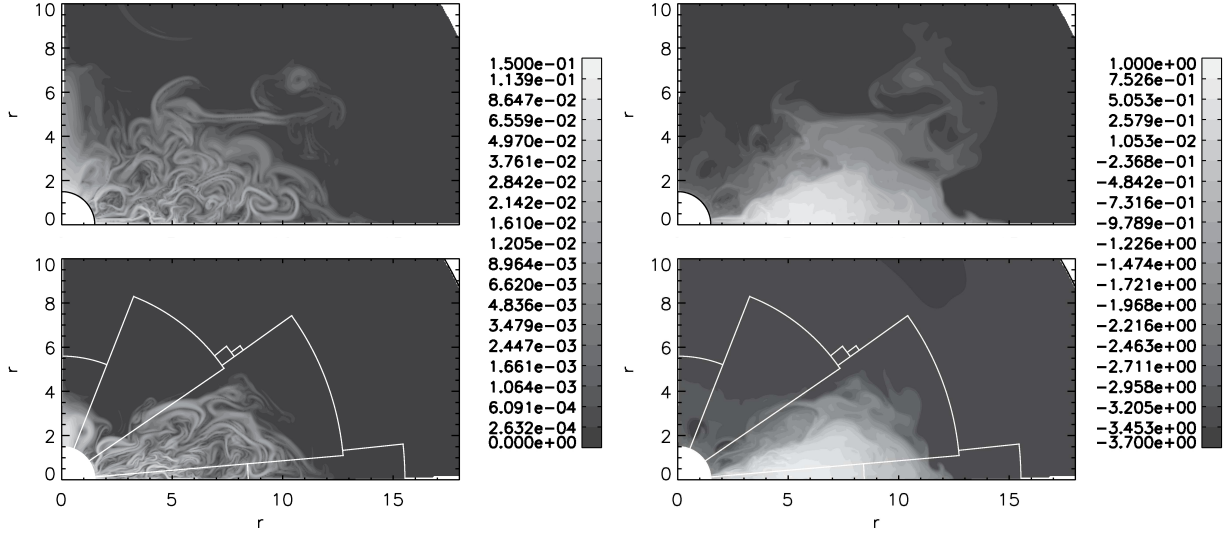


Fig. 13.— The “beard of AZEuS”: Results of MHD accretion torus simulations in (r, ϑ) coordinates at $t = 300$, where the vertical axis is the rotation axis. Plotted are contours of poloidal magnetic field (left) and logarithmic density (right). Top panels are for the uniform grid, bottom panels the AMR solution with one level of refinement. Borders of the adaptive grids are shown with white lines.

$$\begin{aligned}\mathcal{E}_2(r < r_{\text{in}}) &= 2\mathcal{E}_2(r_{\text{in}}) - \mathcal{E}_2(2r_{\text{in}} - r); \\ \mathcal{E}_3(r < r_{\text{in}}) &= 2\mathcal{E}_3(r_{\text{in}}) - \mathcal{E}_3(2r_{\text{in}} - r).\end{aligned}\tag{33}$$

Values of $\mathcal{E}_2(r_{\text{in}})$ and $\mathcal{E}_3(r_{\text{in}})$ are determined by the CMoC algorithm in AZEuS.

Both the uniform grid and AMR solutions are presented here. For the single grid calculation, we use 592 uniform radial zones and 256 uniform meridional zones. For the AMR solution, we use a base grid of 296×128 zones, and allow one level of refinement with a refinement ratio $\nu = 2$, giving the same effective resolution as the uniform grid calculation.

For both calculations, $\mathcal{C} = 0.5$, $\text{qcon} = 1.0$, and $\text{qlin} = 0.1$. For AMR, $\text{kcheck} = 50$, $\text{geffcy} = 0.8$, $\text{ibuff} = 2$, $\text{tolshk} = \text{tolcd} = 0.2$, and $\text{tolgrad} = 0.2$ is applied to B_ϑ . We use the gradient detector to refine on the initial magnetic field configuration (contained entirely within the torus), and the complex field structure that develops later on from the magneto-rotational instability (MRI; Balbus & Hawley 1991), since neither are well-tracked by the shock and CD detectors.

Following Hawley (2000), we enforce a density floor of $10^{-3}\rho(r_{\text{in}})$ to prevent the time step from becoming prohibitively small, and we use the internal energy equation to avoid

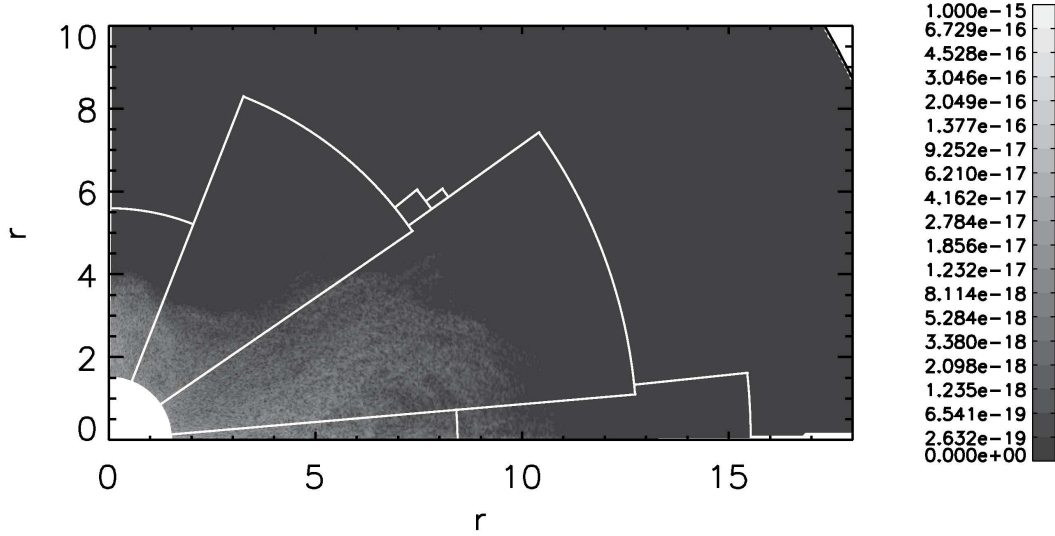


Fig. 14.— AZEuS results for the normalised magnetic field divergence, $|\nabla \cdot \vec{B}| \Delta x / |\vec{B}|$, in the MHD accretion torus simulation at $t = 300$. The maximum value of the normalised divergence at this time, including both physical and adjacent boundary zones, is 1.34×10^{-15} . Borders of the adaptive grids are shown with white lines.

negative pressures. Unlike Hawley (2000), we do not introduce any perturbations on the initial conditions, which slows but does not prevent the onset of MRI.

Figure 13 presents the results of our simulations at $t = 300$. Evidently and very much unlike the Orszag-Tang vortex, the AMR and uniform grid solutions are visually different, even though they have the same effective resolution. For a pseudo-turbulent, “irreversible” problem such as this, small differences between values in the coarse and fine grids grow exponentially during the simulation, and lead to visually different solutions. This is quite unlike the behaviour of a “reversible” problem such as the Orszag-Tang vortex, for which small fluctuations grow at worse linearly in time, and never manifest as visual differences in the plots. On the other hand, integrated quantities tend to be in better agreement than their detailed distributions. For example, between the uniform grid and AMR solutions, the integrated mass and kinetic energy at $t = 300$ differ by only 0.9% and 6.5% respectively. Furthermore, our results are qualitatively similar to Mignone *et al.* (2007); Figure 13 could easily fit in with their Figure 8 for different Riemann solvers and interpolation schemes.

Figure 14 shows the normalised magnetic field divergence, $|\nabla \cdot \vec{B}| \Delta x / |\vec{B}|$, at the simulation end time, where the maximum value is 1.34×10^{-15} , including both physical and adjacent

boundary zones. Indeed, the minimum and maximum values for the divergence throughout the simulation life time reach -7.392×10^{-15} and 7.396×10^{-15} respectively, demonstrating that the algorithms employed for prolongation and restriction of the magnetic field in AZEuS are capable of maintaining the solenoidal condition to machine round-off, assuming the initial conditions are also divergence-free. It is also worthwhile to note the absence of grid boundary effects in Figure 14, which one might expect if monopoles existed within the boundaries of the fine grids.

7.3. A 3-D test: self-gravitational hydrodynamical collapse

The last test is the self-gravitational hydrodynamical collapse calculation first presented by Truelove *et al.* (1997) and Truelove *et al.* (1998; T98). At the time of this writing, self-gravity has not been fully implemented in AZEuS, and so we use the successive over-relaxation (SOR) method (Press *et al.*, 1992) as an easy-to-program, albeit slow, stop-gap measure. Since SOR is incompatible with periodic boundary conditions, we apply inflow boundary conditions instead. This results in some subtle differences from T98, which we discuss below.

A rotating, uniform cloud of mass $M = 1M_{\odot}$ and radius $R = 5 \times 10^{16}$ cm is initialised in the centre of a 3-D Cartesian box with side length $4R$. We use a nearly isothermal equation of state ($p = \kappa\rho^{\gamma}$, $\gamma = 1.001$), initially uniform rotation $\Omega = 7.14 \times 10^{-13}$ rad s $^{-1}$ with angular momentum axis in the positive x_3 -direction, and energy ratios of

$$\alpha = \frac{5}{2} \left(\frac{3}{4\pi\rho_0 M^2} \right)^{1/3} \frac{c_s^2}{G} = 0.16 \quad \text{and} \quad \beta_{\Omega} = \frac{1}{4\pi} \frac{\Omega^2}{G\rho_0} = 0.26, \quad (34)$$

where ρ_0 and c_s are, respectively, the initial density and sound speed of the uniform cloud (T98). The remainder of the computational volume is initialised with uniform density and pressure given by $\rho_{\text{atm}} = 100\rho_{\text{cloud}}$ and $p_{\text{atm}} = 10p_{\text{cloud}}$. Upon the initial uniform density distribution, we apply an azimuthal $m = 2$ perturbation of the form $\rho_{\text{cloud}}(1 + A\cos(2\varphi))$, where $A = 10\%$ is the perturbation amplitude and φ is the azimuthal angle relative to the cloud centre.

Following T98, we set the coarsest grid to 32^3 zones and designate it R_8 (meaning the cloud radius is resolved with eight zones). We immediately add 1 level of refinement by flagging all zones which have density $\rho > \rho_{\text{cloud}}$ (for a refinement ratio of 4, this gives 32 zones per cloud radius, or R_{32}). Beyond this, we enforce the so-called ‘‘Truelove criterion’’:

$$J = \frac{\Delta x}{\lambda_J} = \Delta x \left(\frac{G\rho}{\pi c_s^2} \right)^{1/2} < J_{\text{max}}, \quad (35)$$

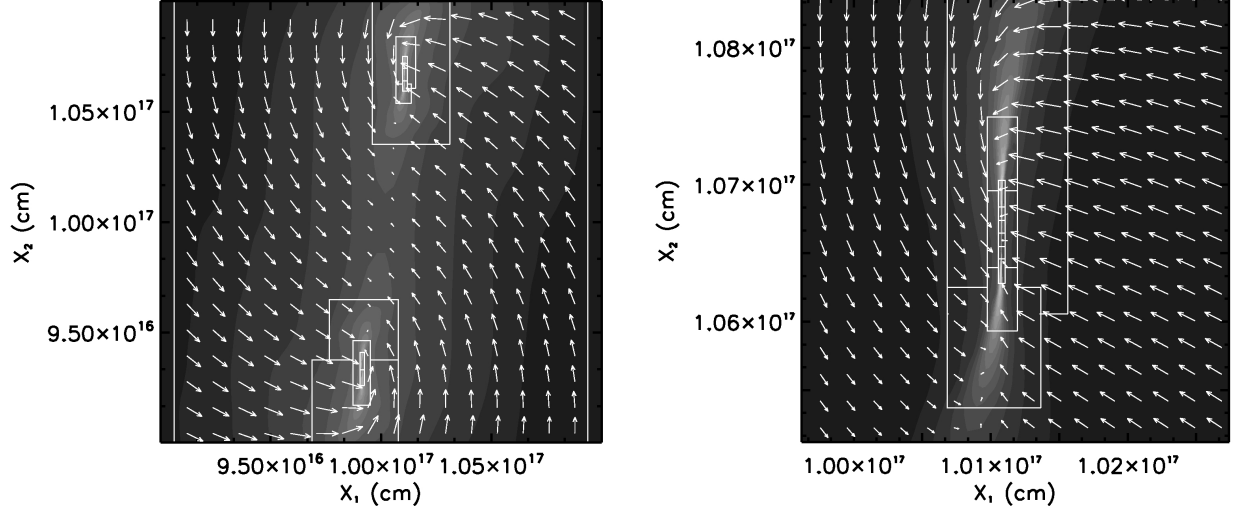


Fig. 15.— AZEuS results for the Truelove problem with a 10% amplitude perturbation at $t = 0.59808 = 1.2152t_{\text{ff}}$, where t_{ff} is the free-fall time. *Left:* Equatorial x_1 - x_2 slice of logarithmic density with velocity vectors superposed. The highest resolution shown here is R_{8192} , five levels of refinement above the base grid. *Right:* Equatorial x_1 - x_2 slice of logarithmic density of the upper fragment. The highest resolution in this plot is R_{32768} (six levels) with one additional level of refinement not shown. White lines denote the borders of AMR grids and the units of density are g cm^{-3} . For each panel, 20 evenly spaced contours are plotted with ranges $\log \rho = [-16.10, -9.905]$ (left) and $[-14.30, -9.604]$ (right).

refining zones which have Jeans numbers J larger than critical value $J_{\text{max}} = 0.25$. Additional run-time parameters for this test include $\mathcal{C} = 0.33$, $\text{qcon} = 1.0$, $\text{qlin} = 0.1$, $\text{kcheck} = 20$, $\text{geffcy} = 0.7$, $\text{ibuff} = 2$, and refinement ratio $\nu = 4$.

Figure 15 shows our results at $t = 0.59808 = 1.2152t_{\text{ff}}$. At this time, there are seven levels of refinement above the base grid (R_{131072}). Our maximum density at this time is $\log \rho_{\text{max}} = -9.347399$, with density measured in g cm^{-3} , an increase over ρ_0 of more than 8 orders of magnitude. While our simulation collapses more quickly than T98 and our Figure 15 does not correspond exactly with their Figures 12 and 13, we also reach a somewhat lower maximum density.

The differences between our results and T98 are most likely caused by the differing boundary conditions (ours inflow, theirs periodic). Indeed, T98 allude to this possibility, suggesting that non-periodic boundary conditions could slightly increase the rate of collapse,

which we observe.

8. Summary

We have described a method for block-based AMR on a fully-staggered mesh, and implemented this method in a new version of *ZEUS-3D* called **AZEuS**. In addition to describing the modifications required to AMR to account for the fully staggered grid, we also describe higher order interpolation methods for the prolongation step which we found necessary to allow for static grids. Static grids are important for problems which, at first order, have a self-similar character and expand over the course of the simulation to ever larger scale lengths. Such a simulation by **AZEuS** has already appeared in the literature ([Ramsey & Clarke, 2011](#)), which showcases the ability of the code to transmit waves of all types and strengths across grid boundaries, and to do so in cylindrical coordinates. The higher-order prolongation operator is designed to maintain the conservation of all important physical quantities such as mass, momentum, energy, and magnetic flux.

Numerous test problems were also presented in 1-, 2-, and 3-D, and in both Cartesian and spherical polar coordinates. These tests demonstrate the ability of the code to produce essentially identical results in “reversible” (non-turbulent) problems whether using a single grid or AMR, and give an example of the differences that can occur in “irreversible” (turbulent) problems as minute differences caused by the insertion or deletion of a grid amplify.

Finally, the **AZEuS** website <http://www.ica.smu.ca/azeus> was introduced on which test problems and simulations will be posted as they become available, and from which the code can be downloaded in the near future.

We thank the referee for a careful reading of the manuscript and for pointing out an oversight in the algorithm as described in the original draft. We also thank Marsha Berger for providing her AMR subroutines. This work is supported, in part, by the Natural Sciences and Engineering Research Council (NSERC). Computing resources were provided by ACEnet, which is funded by CFI, ACOA, and the provinces of Nova Scotia, Newfoundland & Labrador, and New Brunswick.

A. Curvilinear Coordinates

Like other codes in the *ZEUS* family, **AZEuS** uses the “covariant” (coordinate independent) form of the MHD equations ([Stone & Norman, 1992a](#)). Supported geometries in **AZEuS**

(x_1, x_2, x_3)	$g_{12}(j)$	$g_{13}(k)$	$g_{23}(k)$	$g_{21}(i)$	$g_{31}(i)$	$g_{32}(j)$
(x, y, z)	1	1	1	1	1	1
(z, r, φ)	1	1	1	1	1	$x_2(j)$
(r, ϑ, φ)	1	1	1	$x_1(i)$	$x_1(i)$	$\sin x_2(j)$

Table 3: Metric scaling factors in **AZEuS** for Cartesian, cylindrical, and polar coordinates.

include Cartesian (x, y, z) , cylindrical (z, r, φ) , and spherical polar (r, ϑ, φ) coordinates while other orthogonal curvilinear coordinate systems may be easily implemented as needed. For simplicity, the equations presented in the main body of the paper were all written assuming Cartesian-like coordinates. In this appendix, we show how these may be rewritten in a covariant fashion.

The distance differential in an orthogonal coordinate system, (x_1, x_2, x_3) , is given by:

$$ds^2 = g_1^2 dx_1^2 + g_2^2 dx_2^2 + g_3^2 dx_3^2, \quad (\text{A1})$$

where g_i , $i = 1, 2, 3$, are the usual metric scaling factors, all functions of the coordinates. If the scaling factors are separable and independent of their own direction, we can write:

$$\begin{aligned} g_1 &= g_1(j, k) = g_{12}(j) g_{13}(k); \\ g_2 &= g_2(k, i) = g_{23}(k) g_{21}(i); \\ g_3 &= g_3(i, j) = g_{31}(i) g_{32}(j), \end{aligned} \quad (\text{A2})$$

listed in Table 3 for Cartesian, cylindrical, and spherical polar coordinates.

Face areas and zone volumes important for the calculation of fluxes and conserved quantities are written as:

$$\begin{aligned} \delta A_1(i, j, k) &= \int_{x_2(j)}^{x_2(j+1)} \int_{x_3(k)}^{x_3(k+1)} g_{21}(i) g_{31}(i) g_{32}(j') dx'_2 dx'_3 \\ &= g_{21}(i) g_{31}(i) \int_{x_2(j)}^{x_2(j+1)} g_{32}(j') dx'_2 \int_{x_3(k)}^{x_3(k+1)} dx'_3 \\ &\equiv g_{21}(i) g_{31}(i) \delta A_{12}(j) \delta A_{13}(k), \end{aligned} \quad (\text{A3})$$

$$\begin{aligned} \delta A_2(i, j, k) &= \int_{x_3(k)}^{x_3(k+1)} \int_{x_1(i)}^{x_1(i+1)} g_{31}(i') g_{32}(j) dx'_3 dx'_1 \\ &= g_{32}(j) \int_{x_3(k)}^{x_3(k+1)} dx'_3 \int_{x_1(i)}^{x_1(i+1)} g_{31}(i') dx'_1 \end{aligned} \quad (\text{A4})$$

$$\begin{aligned}
&\equiv g_{32}(j) \delta A_{23}(k) \delta A_{21}(i), \\
\delta A_3(i, j, k) &= \int_{x_1(i)}^{x_1(i+1)} \int_{x_2(j)}^{x_2(j+1)} g_{21}(i') dx'_1 dx'_2 \\
&= \int_{x_1(i)}^{x_1(i+1)} g_{21}(i') dx'_1 \int_{x_2(j)}^{x_2(j+1)} dx'_2 \\
&\equiv \delta A_{31}(i) \delta A_{32}(j).
\end{aligned} \tag{A5}$$

$$\begin{aligned}
\delta V(i, j, k) &= \int_{x_1(i)}^{x_1(i+1)} \int_{x_2(j)}^{x_2(j+1)} \int_{x_3(k)}^{x_3(k+1)} g_{21}(i') g_{31}(i') g_{32}(j') dx'_1 dx'_2 dx'_3 \\
&= \int_{x_1(i)}^{x_1(i+1)} g_{21}(i') g_{31}(i') dx'_1 \int_{x_2(j)}^{x_2(j+1)} g_{32}(j') dx'_2 \int_{x_3(k)}^{x_3(k+1)} dx'_3
\end{aligned} \tag{A6}$$

$$\begin{aligned}
&\equiv \int_{x_1(i)}^{x_1(i+1)} dV'_1 \int_{x_2(j)}^{x_2(j+1)} dV'_2 \int_{x_3(k)}^{x_3(k+1)} dV'_3 \\
&\equiv \delta V_1(i) \delta V_2(j) \delta V_3(k),
\end{aligned} \tag{A7}$$

where metric scaling factors equal to 1 in Table 3 have been and will continue to be dropped.

Finally, *ZEUS* has traditionally defined the momentum densities, \vec{s} , as:

$$\begin{aligned}
s_1(i, j, k) &= \rho(i - \tfrac{1}{2}, j, k) v_1(i, j, k); \\
s_2(i, j, k) &= g_{21}(i) \rho(i, j - \tfrac{1}{2}, k) v_2(i, j, k); \\
s_3(i, j, k) &= g_{31}(i) g_{32}(j) \rho(i, j, k - \tfrac{1}{2}) v_3(i, j, k),
\end{aligned} \tag{A8}$$

where the half indices in the density indicate two-point averages. Defining the momentum components with the metric scaling factors simplifies the momentum equation somewhat by eliminating all “Coriolis-like” fictitious forces, leaving only the “centrifugal-like” terms.

A.1. Restriction

A.1.1. Conservative Overwrite

When applying the conservative overwriting procedure on a curvilinear grid, the equations of Section 3.1 must be modified to account for the non-constant volumes of zones. For example, equation (6) for zone-centred quantities becomes:

$$Q(I, J, K) \Delta V_1(I) \Delta V_2(J) \Delta V_3(K) = \sum_{\alpha, \beta, \eta=0}^{\nu-1} q(i + \alpha, j + \beta, k + \eta) \tag{A9}$$

$$\times \delta V_1(i + \alpha) \delta V_2(j + \beta) \delta V_3(k + \eta),$$

and equation (7) for face-centred momenta generalises to:

$$S_1(I, J, K) \Delta V_1(I) \Delta V_2(J) \Delta V_3(K) = \sum_{\alpha=-\nu/2}^{\nu/2} \sum_{\beta, \eta=0}^{\nu-1} \mathcal{G}'(\alpha) s_1(i + \alpha, j + \beta, k + \eta) \quad (\text{A10})$$

$$\times \delta V_1(i + \alpha) \delta V_2(j + \beta) \delta V_3(k + \eta),$$

where:

$$\mathcal{G}'(\alpha) = \begin{cases} w_{1,R}(i + \alpha) & \text{if } \alpha = -\nu/2; \\ w_{1,L}(i + \alpha) & \text{if } \alpha = +\nu/2; \\ 1 & \text{otherwise,} \end{cases} \quad (\text{A11})$$

and where:

$$w_{1,L}(i) = \frac{1}{\delta V_1(i)} \int_{x_1(i-1/2)}^{x_1(i)} dV_1; \quad w_{1,R}(i) = \frac{1}{\delta V_1(i)} \int_{x_1(i)}^{x_1(i+1/2)} dV_1. \quad (\text{A12})$$

While it is still true that the fine zones at $\alpha = \pm\nu/2$ are halfway outside the coarse ROI in *position* space (Figure 3b), this is not generally true in volume space. As an example, consider spherical coordinates where the 1-volume differential is $dV_1(i) = g_{21}(i)g_{31}(i) dx_1(i) = x_1^2(i) dx_1(i)$. Clearly, $dV_1(i)$ in spherical coordinates is not a linear function of position, and it cannot be assumed that exactly 1/2 of the fine momentum volume is outside the coarse ROI. To correct for this, we calculate the ratio of the *half*-volume inside the coarse ROI to the actual volume element (equation A12), which is then used as a weighting factor in $\mathcal{G}'(\alpha)$. For reasonable grid parameters, these weighting factors are small corrections (< a few %) relative to the Cartesian factor of 1/2, but nonetheless are included for accuracy.

Suitable expressions for the 2-direction are obtained by a permutation of indices. No corrections are necessary for the 3-direction, since $dV_3(k) = dx_3(k)$ for the curvilinear coordinates discussed here.

For the magnetic field, equation (9) changes to account for non-uniform areas:

$$B_1(I, J, K) \Delta A_1(I, J, K) = \sum_{\beta, \eta=0}^{\nu-1} b_1(i, j + \beta, k + \eta) \delta A_1(i, j + \beta, k + \eta). \quad (\text{A13})$$

A.1.2. Flux Corrections

Fluxes and EMFs as stored by *ZEUS-3D* already include the appropriate area and length elements, so the required changes for flux corrections in curvilinear coordinates are

not substantial. As with the conservative overwrite, the corrections (*e.g.*, equation 12) need to be adjusted by replacing occurrences of \mathcal{G} (equation 8) with \mathcal{G}' (equation A11). Similar to equation (A10), equation (11) for the flux corrections of momentum components normal to the boundary must also be modified to account for non-constant volumes:

$$\begin{aligned} \tilde{S}_1^{N+1}(I, J, K) = & S_1^{N+1}(I, J, K) - \frac{1}{\Delta V(I, J, K)} \left[F_{1,S_1}^{N+\frac{1}{2}}(I-1, J, K) \right. \\ & - \sum_{\beta, \eta, \tau=0}^{\nu-1} \left(w_{1,R} \left(i - \frac{\nu}{2} \right) f_{1,s_1}^{n+\tau+\frac{1}{2}} \left(i - \frac{\nu}{2} - 1, j + \beta, k + \eta \right) \right. \\ & \left. \left. + w_{1,L} \left(i - \frac{\nu}{2} \right) f_{1,s_1}^{n+\tau+\frac{1}{2}} \left(i - \frac{\nu}{2}, j + \beta, k + \eta \right) \right) \right]. \end{aligned} \quad (\text{A14})$$

By the same argument, equation (13) for coarse momenta with flux components parallel to a grid boundary also needs to be modified, as the original factor of $(\nu-1)/2\nu \equiv \mathcal{R}$ is not correct for curvilinear coordinates:

$$\begin{aligned} \tilde{S}_2^{N+1}(I, J, K) = & S_2^{N+1}(I, J, K) \\ & - \frac{1}{\Delta V(I, J, K)} \left[\mathcal{L}(J, \nu) \left(F_{1,S_2}^{N+\frac{1}{2}}(I, J, K) - F_{1,S_2}^{N+\frac{1}{2}}(I+1, J, K) \right) \right. \\ & \left. - \sum_{\beta=1}^{\nu/2} \sum_{\eta, \tau=0}^{\nu-1} \mathcal{G}'(\beta) \left(f_{1,s_2}^{n+\tau+\frac{1}{2}}(i, j + \beta, k + \eta) - f_{1,s_2}^{n+\tau+\frac{1}{2}}(i + \nu, j + \beta, k + \eta) \right) \right]. \end{aligned} \quad (\text{A15})$$

Here, $\mathcal{L}(J, \nu) = W_{2,L}(J, \nu)$ if $S_2(I, J, K)$ is at the upper edge of the fine grid (high j), and $\mathcal{L}(J, \nu) = W_{2,R}(J, \nu)$ if $S_2(I, J, K)$ is at the lower edge of the fine grid (low j), with:

$$\begin{aligned} W_{2,L}(J, \nu) &= \frac{1}{\Delta V_2(J)} \int_{x_2(J-1/2)}^{x_2(J-1/2+\mathcal{R})} dV_2; \\ W_{2,R}(J, \nu) &= \frac{1}{\Delta V_2(J)} \int_{x_2(J+1/2-\mathcal{R})}^{x_2(J+1/2)} dV_2. \end{aligned}$$

Like the function \mathcal{G}' , \mathcal{L} accounts for the portion of the coarse zone flux in volume space which is being replaced by fine fluxes. As before, expressions for the 2-direction are obtained by a suitable permutations of indices.

A.2. Prolongation

The changes required for prolongation on curvilinear grids deal entirely with making the interpolations conservative for curvilinear volumes and areas. In the case of 1-D PPI,

equation (16) summarising the conservation constraint is modified to:

$$Q(I, J, K) \Delta V_1(I) = \sum_{\alpha=0}^{\nu-1} q(i + \alpha, j, k) \delta V_1(i), \quad (\text{A16})$$

which then affects the rest of the scheme (equations 17 and 18):

$$q_1^*(i + \alpha) = Q_L(I) + \zeta \left(Q_R(I) - Q_L(I) + \mathcal{H}'_1(1 - \zeta) \right), \quad (\text{A17})$$

where:

$$\begin{aligned} \zeta &= \frac{x_1(i + \alpha) - x_1(I)}{\Delta x_1(I)}; \\ \mathcal{H}'_1 &= \frac{1}{f'_1(\nu, i) - f'_2(\nu, i)} \left(\Delta V_1(I) (Q(I, J, K) - Q_L(I)) - f'_1(\nu, i) (Q_R(I) - Q_L(I)) \right); \\ f'_1(\nu, i) &= \frac{1}{2\nu} \sum_{\xi=1}^{\nu} (2\xi - 1) \delta V_1(i + \xi - 1); \\ f'_2(\nu, i) &= \frac{1}{4\nu^2} \sum_{\xi=1}^{\nu} (2\xi - 1)^2 \delta V_1(i + \xi - 1). \end{aligned}$$

As for PLI, the original 1-D scheme in the 1-direction takes the form (van Leer, 1977):

$$q_1^*(i + \alpha) = Q(I) + (2\zeta - 1) \Delta Q', \quad (\text{A18})$$

where:

$$\Delta Q' = \begin{cases} \frac{\Delta Q_R \Delta Q_L}{\Delta Q_R + \Delta Q_L} & \text{if } \Delta Q_R \Delta Q_L > 0; \\ 0, & \text{otherwise,} \end{cases}$$

and where:

$$\Delta Q_R = Q(I + 1) - Q(I); \quad \Delta Q_L = Q(I) - Q(I - 1).$$

Equation (A18) will not generally conserve mass, momentum, or magnetic flux on a curvilinear grid. To correct this, we relax the constraint that the linear interpolation profile must pass through $Q(I)$ at the zone-centre. This releases one degree of freedom that can then be used with equation (A16), resulting in:

$$q_1^*(i + \alpha) = Q(I) + \Delta Q' \left(2\zeta - 1 - \frac{1}{\Delta V_1(I)} f'_3(\nu, i) \right), \quad (\text{A19})$$

where:

$$f'_3(\nu, i) = \frac{1}{\nu} \sum_{\xi=1}^{\nu} ((2\xi - 1) - \nu) \delta V_1(i + \xi - 1).$$

The additional term in equation (A19) can be viewed as either a shift in the zone-centred intercept, or as choosing a modified value of ζ corresponding to the volume-centred rather than the spatial-centred coordinate.

Unfortunately, this “shift” can push $q_1^*(i + \alpha)$ beyond neighbouring values of $Q(I)$ at the edges of the interpolation profile, resulting in a loss of monotonicity. However, since $\zeta = 0$ or 1 at the left or right side of the interpolation profile, we can write:

$$\begin{aligned} \Delta Q'_L &= Q(I) - q_1^*(i + \alpha) = \Delta Q'(1 + \mathcal{K}'_1) \\ \text{or } \Delta Q'_R &= q_1^*(i + \alpha) - Q(I) = \Delta Q'(1 - \mathcal{K}'_1), \end{aligned} \quad (\text{A20})$$

where $\mathcal{K}'_1 = f'_3(\nu, i) / \Delta V_1(I)$. If we limit the slope of the interpolation profile to:

$$\Delta Q'' = \begin{cases} \text{sign}(\Delta Q') |\Delta Q_L| / (1 + \mathcal{K}_1) & \text{if } |\Delta Q'_L| > |\Delta Q_L| \\ \text{sign}(\Delta Q') |\Delta Q_R| / (1 - \mathcal{K}_1) & \text{if } |\Delta Q'_R| > |\Delta Q_R| \\ \Delta Q' & \text{otherwise,} \end{cases} \quad (\text{A21})$$

and, because of the properties of the PLI slope $\Delta Q'$, the first two cases of equation (A21) will never occur simultaneously. Thus, the equation for conservative, monotonic, 1-D PLI in curvilinear coordinates is:

$$q_1^*(i + \alpha) = Q(I) + \Delta Q''((2\zeta - 1) - \mathcal{K}_1). \quad (\text{A22})$$

The previous discussion applies to interpolation of momenta in the directions perpendicular to the component normal (*e.g.*, s_1 in the 2- and 3-directions). The covariant procedure for linear interpolation in between coarse faces (*e.g.*, s_1 in the 1-direction; equation 24) is as follows:

$$\begin{aligned} s_1(i + \alpha, j + \beta, k + \eta) \delta V_1(i + \alpha) &= (1 - \zeta) s_1(i, j + \beta, k + \eta) \delta V_1(i) \\ &\quad + \zeta s_1(i + \nu, j + \beta, k + \eta) \delta V_1(i + \nu). \end{aligned} \quad (\text{A23})$$

Finally, the generalised Li & Li (2004) algorithm is adapted to curvilinear coordinates simply by replacing \vec{B} (and \vec{b}) in equations (25)–(28) with the magnetic “flux functions”:

$$\vec{\Psi} = (\Psi_1, \Psi_2, \Psi_3) = \left(g_{21} g_{31} \frac{\Delta V_2}{\Delta x_2} B_1, g_{31} g_{32} B_2, g_{21} B_3 \right). \quad (\text{A24})$$

With this modification, the solenoidal condition can be written in “Cartesian-like” form, regardless of the geometry:

$$\nabla \cdot \vec{\Psi} = \frac{\partial \Psi_1}{\partial x_1} + \frac{\partial \Psi_2}{\partial x_2} + \frac{\partial \Psi_3}{\partial x_3} = 0,$$

and the prolongation of magnetic field in curvilinear coordinates proceeds exactly as described in Section 4.1.

B. The Vector Potential

Writing $\vec{B} = \nabla \times \vec{A}$ and assuming 3-symmetry with an appropriate gauge, one can easily show from the induction equation (equation 3) that A_3 obeys an “advection equation”:

$$\frac{\partial A_3}{\partial t} + \vec{v} \cdot \nabla A_3 = 0, \quad (\text{B1})$$

from which the poloidal components of the magnetic field are given by:

$$B_1 = \frac{\partial A_3}{\partial x_2}; \quad B_2 = -\frac{\partial A_3}{\partial x_1}. \quad (\text{B2})$$

Further, one can show that B_3 is given by:

$$\frac{\partial B_3}{\partial t} + \nabla \cdot (B_3 \vec{v}) = \vec{B} \cdot \nabla v_3. \quad (\text{B3})$$

Since equation (B3) has exactly the same form as the internal energy equation (equation 4), and since equation (B1) is a simple advection equation, the induction step in the original *ZEUS* code (zeus04; Clarke 1988) was based on solving these equations using the hydrodynamical algorithms already in the code, and then calculating B_1 and B_2 from equations (B2). Evidently, the solenoid condition is *strictly* satisfied. Regardless of the initial magnetic field configuration, it is easy to show that a face-centred \vec{B} and a corner-centred (edge-centred in 3-D) A_3 guarantees $\nabla \cdot \vec{B} = 0$ everywhere and at all times to machine round-off error.

However, Clarke (1988) points out that a second-order accurate A_3 means first order accurate poloidal magnetic field components and a *zeroth-order* accurate current density J_3 (each differentiation reduces the order of accuracy by one), and this was found to have catastrophic consequences in computing the $\vec{J} \times \vec{B}$ source terms in the momentum equation (equation 2). Thus, the vector potential algorithm in zeus04 was abandoned, and the first publicly released version of *ZEUS-2D* (Stone & Norman 1992a,b), and later *ZEUS-3D* (Clarke 1996), were based on the CT algorithm of Evans & Hawley (1988) in which the magnetic field is updated directly. Note that the CT scheme *conditionally* satisfies the solenoidal condition, requiring that the magnetic field be initialised such that $\nabla \cdot \vec{B} = 0$.

Still, the failure of the vector potential algorithm in zeus04 is not a complete indictment of \vec{A} for use in numerical MHD algorithms. Indeed, Londrillo & Del Zanna (2000) and Igumenshchev & Narayan (2002) have successfully demonstrated the use of \vec{A} as the primary magnetic variable in their MHD codes. By substituting $\vec{B} = \nabla \times \vec{A}$ into equation (3) and with an appropriate gauge, we can write:

$$\frac{\partial \vec{A}}{\partial t} = \vec{v} \times \vec{B} \equiv -\vec{E}. \quad (\text{B4})$$

Thus, CT can be used as originally designed in which $\nabla \times \vec{E}$ is used to update \vec{B} , or easily modified to use \vec{E} to update \vec{A} directly, then update \vec{B} by taking a curl of the updated \vec{A} . Either way, a curl must be taken and the algorithms are interchangeable to machine round-off error. Based as they are on such a modified CT scheme, the vector-potential algorithms used by [Londrillo & Del Zanna \(2000\)](#) and [Igumenshchev & Narayan \(2002\)](#) are very different from the failed algorithm described for **zeus04**, showing none of the effects of the inaccurate current density.

Based on this observation, preliminary versions of **AZEuS** used the vector potential as the primary magnetic variable so that prolongation could be accomplished by interpolating \vec{A} (rather than \vec{B}), thus guaranteeing preservation of the solenoidal condition on a newly-created fine grid or in a fine boundary region. Furthermore, because the vector potential conserves magnetic flux via a path integral ($\oint \vec{A} \cdot d\vec{l} = \int \vec{B} \cdot d\vec{\sigma}$), restricting \vec{A} then calculating \vec{B} means that no EMF corrections are required at adjacent boundaries. This approach, however, was found to be unsatisfactory since the parabolic interpolation function generated by PPI on \vec{A} produces piecewise-linear profiles for \vec{B} and piecewise-constant profiles for \vec{J} , recovering the problem that doomed the original **zeus04** algorithm.

In addition, having to match gauges on overlapping grids poses a significant problem, and one that we never solved. In order to arrive at equation (B4), one implicitly assumes a specific gauge. For a single grid, this is not a problem since there is never a need to specify this gauge. For multiple grids, however, each grid may have its own gauge (especially for grids whose origins are not coincident) and leaving them unspecified gives rise to discontinuities in the perpendicular components of the magnetic field at adjacent boundaries. While a solution to this problem likely exists, we abandoned vector potentials in **AZEuS** before finding one because of the insurmountable problem of the lack of accuracy in the current densities.

As an illustration, Figure 16 shows early results of the simulations in [Ramsey & Clarke \(2011\)](#) in which the vector potential is used as the primary magnetic variable. The left panel shows the solution immediately before a grid modification, and the right panel the solution a few time steps after and after the fine grid was extended from $x_1 = 160$ to $x_1 = 166$. The errors committed by the piecewise constant current densities take a while to dissipate and, in this particular example, result in particularly egregious defects in the velocity divergence distribution within the new portion of the grid. Conversely, the [Li & Li \(2004\)](#) algorithm we currently employ renders adjacent boundaries virtually invisible in the distributions of all variables.

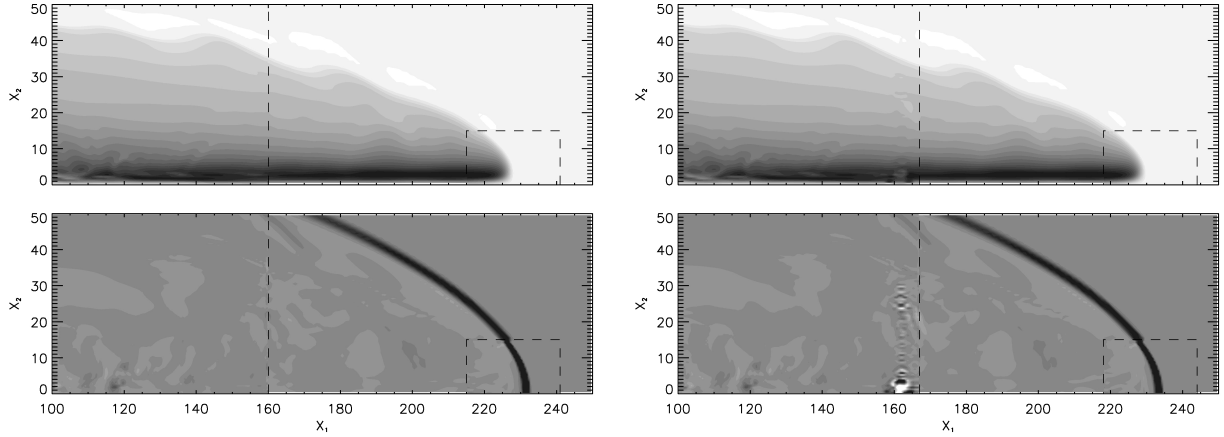


Fig. 16.— The effects of differencing a parabolic interpolation of \vec{A} twice to calculate $\vec{J} \times \vec{B}$ forces. Left panel: the solution immediately before a grid adaptation step. Right panel: the solution a few time steps after. Plotted are 20 evenly spaced contours of the toroidal magnetic field (top) and velocity divergence (bottom) with ranges $B_\varphi = [-0.035, 0.0]$ and $\nabla \cdot \vec{v} = [-0.15, 0.15]$, respectively.

C. The effects of a monopole in the boundary

Since AZEuS is vulnerable to boundary pollution (Section 5), we must take care that values in the self-computed boundaries of AMR grids are as free from non-physical phenomena (*e.g.*, monopoles) as the grid itself. To demonstrate this, Figure 17 presents the results from a simulation with initially uniform and quiescent conditions $(\rho, p, \vec{v}, B_1, B_2, B_3) = (1, 0.6, 0, 0, 1, 0)$, where a static grid with corners $[\pm 0.2, \pm 0.2]$ is positioned in the centre of a Cartesian domain $(x_1, x_2) \in [-0.5, 0.5]$. A single “field defect” ($\vec{B} = 2B_2\hat{x}_2; \nabla \cdot \vec{B} \Delta x_2/B_2 = \pm 1$) is deliberately placed in the left boundary of the static grid at $x_2 = 0$ and $t = 0$. This field defect persists only for a single fine time step before the boundaries are replaced by data prolonged from the underlying coarse values in the manner described in Section 4. Other parameters used in this simulation include a refinement ratio of $\nu = 2$, $\gamma = 5/3$, $\mathcal{C} = 0.5$, $\text{qcon} = 1.0$, and $\text{qlin} = 0.1$ (as defined in Table 1). The base grid has a resolution of 200×200 zones, and all physical boundaries are set to outflow conditions.

Although the field defect is only present for a single fine time step, a fast magnetosonic wave is still launched from the boundary of the fine grid, propagating into both fine and coarse grids. All deviations from quiescence result from the momentary presence of the monopole in the boundary, leading to what is clearly an unacceptable result. And yet, $\nabla \cdot \vec{B} = 0$ to within machine round-off error everywhere on the active grid at all times

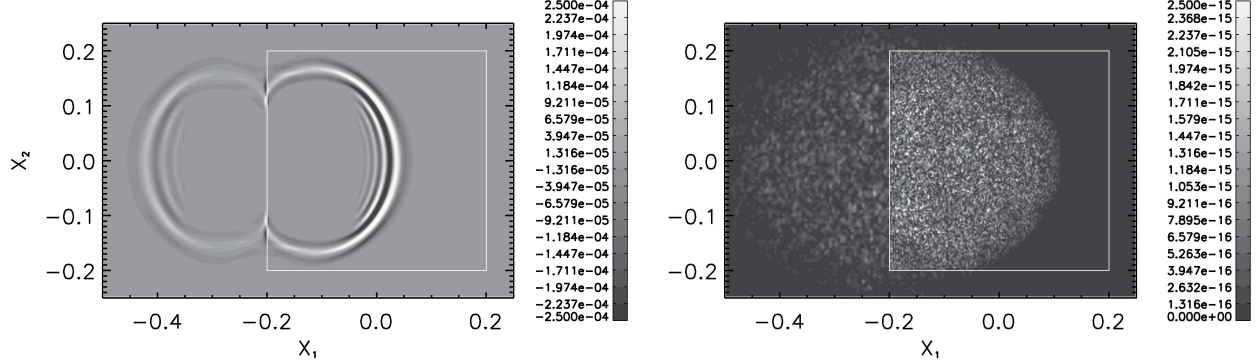


Fig. 17.— The results of placing a “field defect” at $x_2 = 0$ in the left boundary of a static grid for a single fine time step. Plotted are contours of velocity divergence (left) and normalised magnetic field divergence (right). The static grid boundaries are shown as a solid white line, and the simulation is depicted at $t = 0.17$.

(*e.g.*, Figure 17, right), and additionally within the boundaries after the first fine time step. Therefore, the magnetic field prolongation and restriction operators in **AZEuS** are designed to preserve magnetic divergence to machine round-off error in both the active grid and in the boundaries at all times.

D. Schematic overview of the AMR module

This appendix is designed mainly for programmers and those who may wish to use and/or modify **AZEuS**. It is meant as an overview to illustrate how the main ideas covered in this paper have been implemented in the code.

Step 0. **Initialise** computational domain and all variables for the current run.

Set level `lv1 = 1`, `ntogo(lv1) = 1`.

MAIN LOOP:

Step 1. **For** level `lv1`, call `REGRID(lv1)` if `kcheck` cycles have transpired at level `lv1` since the last call to `REGRID`, or if `t = 0`.

For `l = maxlevel - 1, lv1, -1`:

Step 1a. Flag existing grids at level `l` for refinement based on one or more physical criteria. Add `ibuff` buffer zones around flagged points.

- Step 1b. Create grids around flagged points based on **geffcy**.
- Step 1c. Check for proper nesting on all new/modified grids; fix grids which are not properly nested via bisection.
- Step 1d. If any new/modified grids are abutting, make them overlapping instead.

End For

For $l = lvl + 1$, **maxlevel**:

- Step 1e. Fill new/modified grids with values either from old grids at level l or interpolated from coarse grids at level $l - 1$.
- Step 1f. Remove old grids which are no longer in use.

End For

Step 2. **For** level lvl , call **ADVANCE**(lvl , dt).

- Step 2a. **For** all grids at level lvl , fill boundary zones either from overlapping grids at the same level, or interpolate values from coarse grids at level $lvl - 1$ (unless $lvl = 1$, in which case use physical boundary values).

For each grid m at level lvl :

- Step 2b. Advance grid m by time step dt with *ZEUS-3D*.
- Step 2c. Save fluxes/EMFs along the edges of grid m for flux corrections later.

End For

Step 3. One time step at level lvl is complete; check for levels $> lvl$.

Set $ntogo(lvl) = ntogo(lvl) - 1$.

Step 3a. **If** $lvl < maxlevel$ then:

Set $lvl = lvl + 1$.
Set $ntogo(lvl) = nu$, where nu is the refinement ratio.
Set $dt = dt / ntogo(lvl)$.
Go to the beginning of the main loop.

End If

- Step 3b. **If** $ntogo(lvl) > 0$ then **go to** the top of the main loop,
Else Set $lvl = lvl - 1$.

Step 4. **For** level lvl , call **UPDATE**(lvl).

For each grid m at level lvl :

- Step 4a. Flux correct each grid m with fluxes from grids at level $lvl + 1$, if any.

Step 4b. Overwrite zones on grid m with overlying zones at level $lv1 + 1$, if any.

Step 4c. Update any physical (non-periodic) boundary values which depend on zones which were just flux corrected or overwritten.

End For

If $lv1 > 1$, go to Step 3b.

Step 5. One entire AMR cycle is complete. Reconcile time steps across all grids and levels.

Step 6. Perform any required input/output.

Step 7. **If** $t < tlimit$, **go to** the top of the main loop, **else** exit.

REFERENCES

- Balbus, S. A., Hawley, J. F., 1991, ApJ, 376, 214
- Balsara, D. S., 2001, JCoPh, 174, 614
- Bell, J., Berger, M., Saltzman, J., Welcome, M., 1994, SIAM J. Sci. Comput., 15, 127
- Berger, M. J., Oliger, J., 1984, JCoPh, 53, 484
- Berger, M. J., Colella, P., 1989, JCoPh, 82, 64
- Berger, M. J., Rigoutsos, I., 1991, IEEE Trans. Systems Man and Cybernet., 21, 1278
- Clarke, D. A., 1988, Ph.D. dissertation, The University of New Mexico
- Clarke, D. A., 1996, ApJ, 457, 291
- Clarke, D. A., 2007, *ZEUS-3D* User Manual (Version 3.5), <http://www.ica.smu.ca/zeus3d>
- Clarke, D. A., 2010, ApJS, 187, 119
- Colella, P., Woodward, P. R., 1984, JCoPh, 54, 174
- Collins, D. C., Xu, H., Norman, M. L., Li, H., Li, S., 2010, ApJS, 186, 308
- Cunningham, A. J., Frank, A., Varnière, P., Mitran, S., Jones, T. W., 2009, ApJS, 182, 519
- Evans, C. R., Hawley, J. F., 1988, ApJ, 332, 659

- Falle, S. A. E. G., 2002, ApJ, 577, L123
- Fromang, S., Hennebelle, P., Teyssier, R., 2006, A&A, 457, 371
- Fryxell, B., *et al.*, 2000, ApJS, 131, 273
- Gardiner, T. A., Stone, J. M., 2005, JCoPh, 205, 509
- Hawley, J. F., 2000, ApJ, 528, 462
- Hayes, J. C., Norman, M. L., Fiedler, R. A., Bordner, J. O., Li, P. S., Clark, S. E., ud-Doula, A., Mac Low, M.-M., 2006, ApJS, 165, 188
- Igumenshchev, I. V., Narayan, R., 2002, ApJ, 566, 137
- Khokhlov, A. M., 1998, JCoPh, 143, 519
- Klein, R. I., 1999, J. Comput. Appl. Math., 109, 123
- Kritsuk, A. G., *et al.*, 2011, ApJ, 737, 13
- Li, S., Li, H., 2004, JCoPh, 199, 1
- Londrillo, P., Del Zanna, L., 2000, ApJ, 530, 508
- Mignone, A., Bodo, G., Massaglia, S., Matsakos, T., Tesileanu, O., Zanni, C., Ferrari, A., 2007, ApJS, 170, 228
- Orszag, S. A., Tang, C.-M., 1979, J. Fluid Mech., 90, 129
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 1992, Numerical Recipes in FORTRAN: The Art of Scientific Computing (2nd ed.; Cambridge: Cambridge Univ. Press)
- Ramsey, J. P., Clarke, D. A., 2011, ApJ, 728, L11
- Ryu, D., Jones, T. W., 1995, ApJ, 442, 228
- Stone, J. M., Norman, M. L., 1992, ApJS, 80, 753
- Stone, J. M., Norman, M. L., 1992, ApJS, 80, 791
- Stone, J. M., Mihalas, D., Norman, M. L., 1992, ApJS, 80, 819
- Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., Simon, J. B., 2008, ApJS, 178, 137

- Truelove, J. K., Klein, R. I., McKee, C. F., Holliman, J. H., II, Howell, L. H., Greenough, J. A., 1997, ApJ, 489, L179
- Truelove, J. K., Klein, R. I., McKee, C. F., Holliman, J. H., II, Howell, L. H., Greenough, J. A., Woods, D. T., 1998, ApJ, 495, 821
- van Leer, B., 1977, JCoPh, 23, 276
- von Neumann, J., Richtmyer, R. D., 1950, J. Appl. Phys., 21, 232
- Wise, J. H., Abel, T., 2011, MNRAS, 414, 3458
- Ziegler, U., 2008, Comput. Phys. Comm., 179, 227